Magyar számítástechnikai folyóirat CD-melléklettel Ára: 648 Ft

XVII. évfolyam 6. szám 1999. június

ÚJ ALAPLAP

A hónap témája:

KÖRNYEZETVÁLTOZÁS



A DB2 az élre tör

Szoftverportéka

Telepítés, regisztrálás...

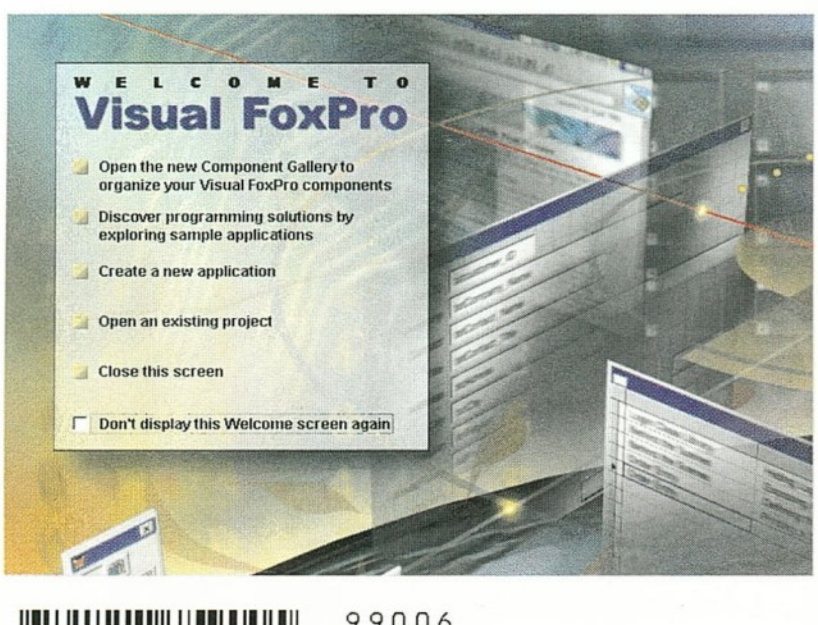
fogódzó

Linux szeminárium Tel Aviv-ban

Alternatíva

A hétfejű terminológiai sárkány

Könyvespolc



99006
9 771217 759005


ElenderMan.
Eléri a célját.
EURO RSCG BUDAPEST

ÚJ ALAPLAP

A Mikroszámítógép Magazin és az Alaplap hagyományait folytató magyar számítástechnikai folyóirat

Megjelenik havonta, CD-melléklettel

Főszerkesztő:
Faklen Pál

Szerkesztő:
Jakab Ágnes

A szerkesztőbizottság tagjai:
Adorjáni Gábor, Ambrózy Gábor, Aszalós László, Bánó György, Feleki Zoltán, Galántai Zoltán, Herczeg József, Kádár Zsolt, Kovács Attila, Mákos András, Nagy Tamás, Pogány Csaba, Protzner György, Simay Endre István, Szappanos Gábor, Szondi Egon János, Vargha Dénes, Vékony Tamás

Szerkesztőség és kiadó:
1539 Budapest, Pf. 571
VI., Dózsa György út 84/b
Telefon: 322-4417, 322-5238
Fax: 351-8015
E-mail: alaplap@mail.datanet.hu

Weblap: http://www.alaplap.hu

Felelős kiadó:
Faklen Pál

Terjesztés:
Megyes Zsuzsanna

Hirdetésszervezés:
Árvai Katalin, Bogácsi Mária, Galyasi Hedvig, Pap Katalin

Külföldi hirdetések:
PubliCity
Reklám- és Médiaügynökség
1537 Budapest I., Márvány u. 17.
Telefon: 356-1182 Fax: 375-3539

A kiadó a hirdetések tartalmáért és a nyomdakészen kapott hirdetések formájáért (és helyesírásáért) nem vállal felelősséget

Példányszámadatok hitelesítése:
Magyar Terjesztésellenőrző Szövetség
MATESZ
Ez a szám
10 000 példányban jelent meg

Nyomtatás:
Zalai Nyomda Rt, Zalaegerszeg
Felelős vezető:
Czirkl György vezérigazgató

Terjeszti:
A Magyar Posta Rt, a Nemzeti Hírlapkereskedelmi Rt, a Hírker Rt, a Kiadói Lapterjesztő Kft és számos számítástechnikai szaküzlet

Előfizethető a kiadónál:
Új Alaplap Kiadói Kft,
1539 Budapest, Pf. 571
Bankszámlaszám:
OTP 11706016-20788599

A lap példányonkénti ára: 648 Ft
Évi előfizetési díj: 6480 Ft

Külföldi előfizetés díja:
6480 Ft + postázási költség

HU ISSN 1217-7598

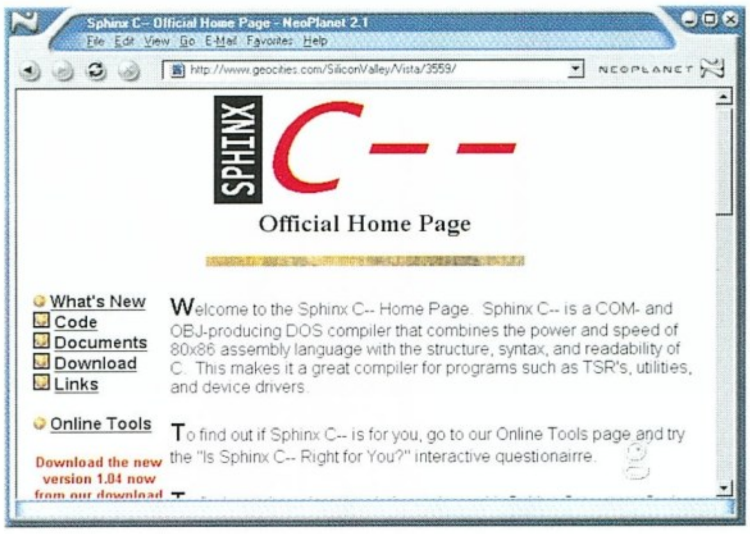
XVII. ÉVFOLYAM 6. SZÁM, 1999. JÚNIUS

TARTALOM

A CD-MELLÉKLETEN

1999. JÚNIUS

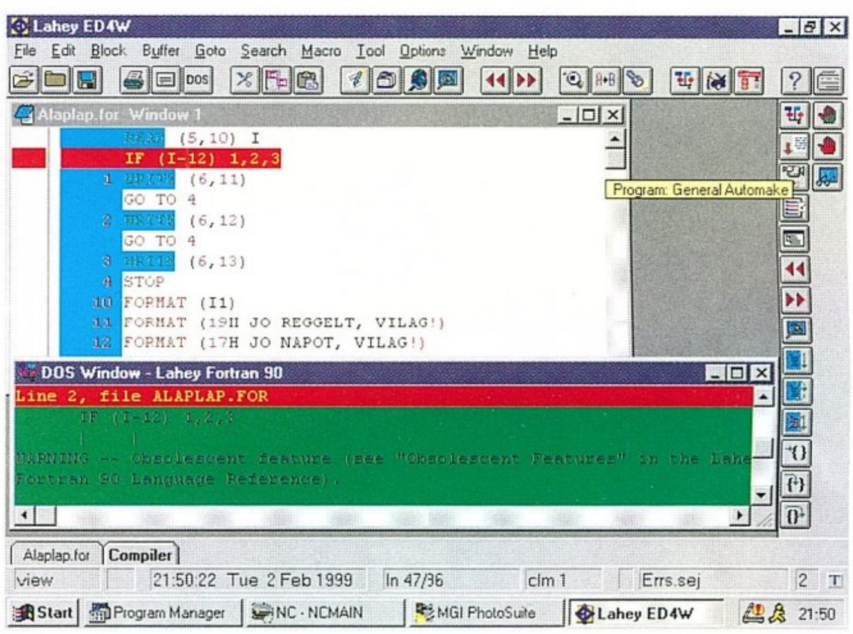




FOKUSZ	***Fókuszban a programozás környezetváltozása***
ASIDE	BlackCat Aside v1.1, a védett módban futó Assembly IDE (shareware)
BASIC	Basic különböző platformokra (Linux, Mac, Windows)
COMPILER	Cikksorozat a compilerek készítéséről
FORTRAN	Fortran fordítók, dokumentációk
FREEPAS	Free Pascal, Pascal fordítók különböző platformokra (DOS, Linux, OS/2, Windows)
GNUC	GNU C fordító (DOS, OS/2, Windows)
IBM	Visual Age for Java v2.0 OS/2-re
JAVA	Java fejlesztői környezetek (Zsemlye Tamás)
JIDE	JIDE v1.7, fejlesztőkeret Sun Java DK 1.1.x-hez
LOGO	Microsoft Windows Logo és Comenius Logo H
MICROSFT	Microsoft Macro Assembler 6.14 Patch a korábbi verziókhoz
NASM	Netwide Assembler v0.97
OBERON	Oberon Linuxra és Windowsra
PERL	A Perl szkriptnyelv
PMAN	Project Manager for Windows általános fejlesztői segédlet Windowsra
PVIDE	Power View IDE v5.0, shareware fejlesztőkörnyezet Watcom C/C++-hoz
PYTHON	Python fejlesztőeszköz
SPHINX	Sphinx C-- v1.06, C-- nyelvi környezet DOS-hoz
ST	ST IDE v1.0, fejlesztőkeret több nyelvhez
SYBASE	PowerJ Learning Edition, a Sybase javás fejlesztőkörnyezetének szabadon hozzáférhető verziója
SZONDI	Fortran illusztrációk (a 64. oldali cikkhez)
TCL_TK	TCL szkriptnyelv
VISAGE	VisualAge Generator (Vásárhelyi Dániel
VISASM	Visual Assembler
LAPFORG	***Lapraforgó***
HERCZEG	Kuk@cológia (27. oldal)
NAGY	Közkincs (40. oldal)
SZERSZAM	***Szerszámosláda***
DOS	DOS-os alkalmazások
WIN3X	16 bites windowsos alkalmazások
WIN9X	32 bites windowsos alkalmazások
LINUX	Linuxos alkalmazások
OS/2	OS/2-es alkalmazások
FESZER	Gyakran szükséges programok
VIRUS	Vírusirtók
VENDEG	***Vendégoldal***
DELCMP	Delphi komponensek
FREEBSD	FreeBSD csomagok
IBM	Java Developer Kit és Java Runtime Environment (v1.1.7)
KKERESO	A WebHu „KincsKereső" offline változata
MICROSFT	Microsoft Java SDK v3.2, DirectX 6.1 SDK
REDHAT	Red Hat 5.2 kiegészítő csomagok
SUN	Sun Java Development Kit frissítések, dokumentációk
JATEKVAR	***Játékvár***
TOTO	Totózóprogram (Gubics Péter)
BLOCK	Tetris 32 bites Windowsra (shareware)
FREECIV	Free Civilization 1.8, a Civilization program szabadon terjeszthető változata
REBUSZ	A PC Rébusz júniusi száma (Nagy Gyula)







Korszakok a programozásban

A kódolók trónfosztása

A szoftverek „születési körülményeivel” foglalkozva régebben többnyire programozókról, programnyelvekről esett szó. Az utóbbi években ehelyett egyre inkább fejlesztőket, fejlesztőkörnyezeteket emlegetünk. Vajon csak szóhasználati divat, valamiféle előkelősdi, vagy egy sokkal mélyebb folyamat megnyilvánulása ez? Piedesztálon van-e még a programozás, a kódolás, vagy már megtörtént a trónfosztás?

A gép előtt ülve valószínűleg kivétel nélkül mindenkivel elő szokott fordulni, hogy elégedetlen az általa használt egyik-másik programmal. Ha pedig a felhasználó egyúttal programozó is, könnyen támad olyan érzése, hogy bizonyos dolgokat ő maga jobban meg tudna oldani. Az alkalmazási feladatok adottak, a megoldásukra hivatott gyári programok számos (slendrián vagy tudatos) hiányosságtól terhesek, nem csoda tehát, hogy ez mindig is programírásra ösztönözte a „kívülállókat”. Ilyen indíttatásból rengeteg szabadon hozzáférhető shareware, freewaree vagy public domain program készült a személyi számítógépek elterjedését követően, és ez a lelkesedés még ma is tart.

Külön informatikai szakma

A személyi számítógépek megjelenése előtt a programok nem válhattak közkinccsé. A programozás kevesek „kiváltsága” volt, a gépidő sokba került. A teremnyi méretű gépeket sokáig csak a nagy matematikai apparátust igénylő feladatok célgépeinek tekintették, majd pedig az ügyviteli adatfeldolgozás „rabszolgáinak”. Ha a programozáshoz és a gép lelkéhez kevésbé értő főnök másra (is) szerette volna használni e monstrumokat, többnyire a gépnek és kezelőjének korlátai bizonyultak erősebbnek.

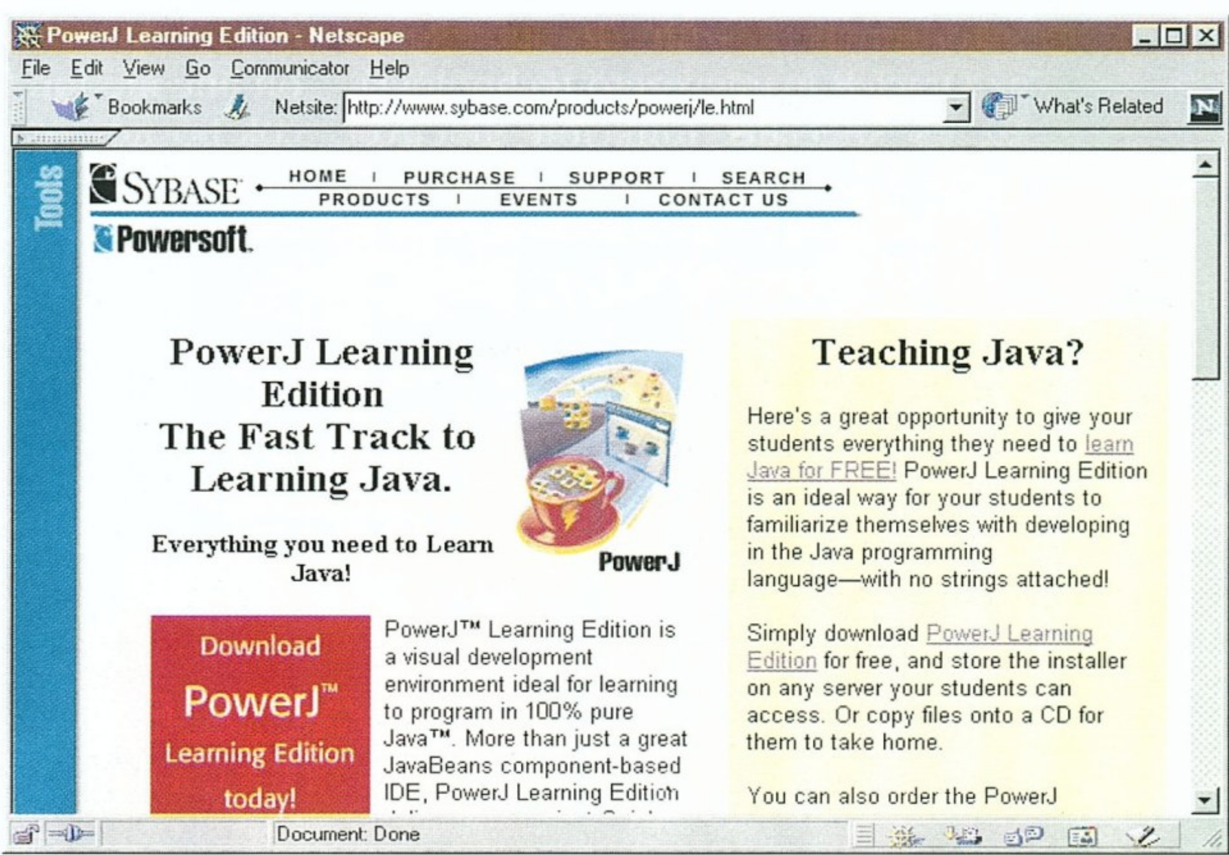
A személyi számítógépek megjelenésével valamilyen szinten nagyon sokan belekóstolhattak a programozásba, és átélhették azt a sikerélményt, amit a kódsorok begépelése után saját készítésű programjuk sikeres elindítása okozott. Egyre bővült azok köre, akik programkódok előállítására szakosodtak. E szakemberek általában jól értettek az általuk használt géphez, és az arra telepített programkörnyezethez, de ez ritkán járt együtt a feladat megoldásának hatékony és komplex megtervezésével. Az ugyanis külön informatikai szakma, egyesek szerint egész tudományág. A nagyüzemi programozók körében egyre jobban elkülönült a „kódolók”, azaz programkódírók csapata, és ezt az elkülönülést a programrendszerek bonyolultságának növekedése, a forráskódkészítés munkamegosztása is elősegítette.

A számítógépes alkalmazásokat készítő programgyárakban szervezetileg és fizikailag is elhatárolódtak a programozás lépései. A munka egészét inkább már programfejlesztésnek nevezték, olyan összetett folyamatnak, amely a feladat megfogalmazásától a megoldást szolgáló alkalmazás megtervezésén át az alkalmazás tényleges elkészítéséig terjed. Az utóbbi fázisba értelemszerűen beletartozik a kód megírása és a gép által értelmezhető nyelvre történő fordítása, majd a program kipróbálása, javítása, optimalizálása — ciklikusan többször ismételve.

Összeállításunk leginkább ez utóbbi témát, a kódkészítést körüljárva igyekszik annak eszköztárát áttekinteni. Más alkalommal sort kerítünk majd arra is, hogy a tervezés kérdéseivel foglalkozzunk, de itt is megemlítjük, hogy ma már a tervezési munkafázisok eredményességének javítására ugyancsak rendelkezésre állnak különböző segédeszközök, modellezhetők a komplex folyamatok, és sok esetben akár a tervezőeszköz is képes az alkalmazás vázát különböző programnyelveken megjelenő kódként előállítani.

Melyik nyelvet válasszuk?

A kódkészítővé válás egyik elkerülhetetlen kérdése, hogy milyen programnyelvet tanuljunk meg. A legkézenfekvőbb válasz az lenne, hogy olyat, amely az adott feladatnak leginkább megfelel. (A merészebbek saját programnyelv megírásával is kacérkodhatnak, hiszen a jelenlegi programnyelvek is hasonlóképpen jöttek létre.) A nagygépes környezetben más-más programnyelv volt az elsősorban matematikai célú, vagy az adatbázisokat feldolgozó programokhoz, a programozás tanulásához, az általános feladatmegoldáshoz, vagy a klasszikus memóriacsata programokhoz. Az Algol, Cobol, Fortran stb. nyelvekhez rendre elkészültek az adott géptípusnak megfelelő fordítóprogramok is, amelyek a szöveges formában megírt programkódból a gép által érhető kódot előállították. Egy-egy feladathoz ké-



szült célgépek esetén pedig egy-egy programozói csapatnak elegendő volt az arra készült fordítóprogram igényeinek megfelelő nyelvjárást ismerni.

A gépek száma azonban rohamosan nőtt, méretük csökkent, a sokasodó feladatok pedig azt követelték meg, hogy a már egyszer megtanult programnyelvet (az egyre bonyolultabbá és hosszabbá váló forráskódokkal együtt) lehetőség szerint a többi környezetben is használni lehessen. Ez az igény hozta létre például a C nyelvet. A különböző programnyelvek kialakulásának folyamata azonban napjainkban is zajlik, ugyanakkor a régiek sem haltak ki teljesen. Így arra a felvetésre, hogy lehet-e még ma is „csináld magad" programnyelveket készíteni, egyértelműen igen a válasz. Mellé kell azonban tenni a program futását lehetővé tevő futtató- és fordítóprogramot is... És még valamit: egy fejlesztőkörnyezetet. Az utóbbi időben a hagyományos programnyelvek felhasználásának elősegítéséhez ebbe az irányba kellett lépéseket tenni.

A program valamikor szép nagy űrlapokon készült, amelyekről a lyukasztást végzők készítettek géppel beolvasható lyukkártyákat, lyukszalagokat. Azután elterjedt a soreditor, amely még nem volt valami „felhasználóbarát", de arra nagyon jónak bizonyult, hogy hozzáértő guruk végezzék a programozást, gondosan átgondolva mindent, amit leírnak.

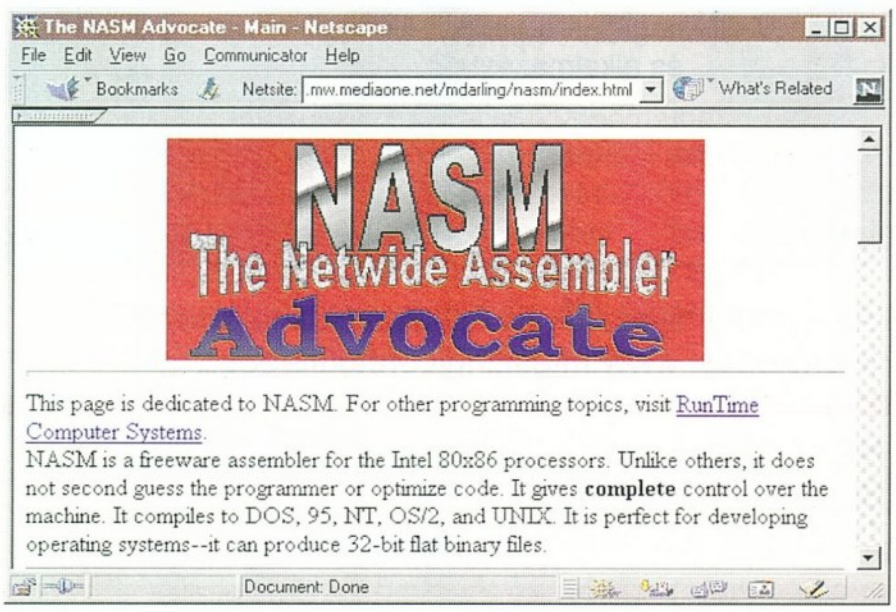
A szövegszerkesztők már a teljes képernyőn tették lehetővé a kód írását. A készülő kódot minden egyes próbafordítás előtt el kellett menteni, majd vagy a fordító, vagy a futás közben elakadt program hibaüzenetei alapján hozzáfogni a hibajavításhoz. A folyamat hatékonyságába bárki betekinthet a jelenlegi PC-ken is, ha megpróbálja a legegyszerűbb DOS-hoz kapott QBasic programot például a Norton Commander beépített editorával előállítani, majd a QBasic keretrendszerének elindítása nélkül közvetlenül futtatni. Ez a keretrendszer ugyanis már meglehetősen komplex fejlesztőkörnyezet, amit a nem is olyan régi C vagy Pascal nyelven programozók is megirigyelhettek. A kódot integráltan lehet szerkeszteni, futtatni, és hiba esetén javítani. Hasonló funkcionalitásúak az integrált fejlesztőkörnyezetek is (Integrated Development Environment — IDE), amelyek ma már az elterjedt programnyelvek kötelező tartozékai, piaci belépőjegyei.

Objektumosodás

A fejlesztőkörnyezetek hosszú ideig csak karakteres felületre, a PC-k esetében általában DOS operációs rendszerre készültek. Még olyan esetekben is, amikor már grafikus felületen kommunikáló programokat készítettek velük. Például a DOS-ból indulva, annak keretein belül valósították meg a grafikus felületet, vagy saját memóriakezeléssel bővítették ki a DOS nem túl nagy mozgásterét. A memóriagazdálkodást olyan nyelvi újítások sem hagyták érintetlenül, mint az objektumorientált programozás megjelenése és térhódítása.

Az objektumorientált programozást támogató nyelvekben a program egyes elemeit — mintegy a strukturált programozás kicsúcsosodásaként — önálló objektumokként határozzák meg. Az objektumoknak saját eljárásaik és változóik vannak, amelyek felhasználási körét a programozó maga szabhatja meg. Egyesek csak az adott objektum számára elérhetők, mások minden leszármazott objektum vagy az objektummal dolgozó más programrész számára is.

De az objektum minden esetben saját zárt világában létező programegység, amelyet a legtöbb programnyelv a változókhoz hasonlóan kezel, beleértve a számukra lefoglalt memóriát is. Ez pedig annál nagyobb, minél több az objektumban definiált változó és eljárás, jóllehet magára az objektumra esetleg csak egyetlen memóriamutatóval hivatkozunk. (Külön szépsége a dolognak, hogy definiálhatók olyan változók és eljárások is, amelyek az objektum valamennyi példányában, beleértve leszármazottjait is, minden esetben értéket kapnak, illetve lefutnak.)



Képzeljünk magunk elé egy objektumhierarchián alapuló programot, amelynek alapja egy téglalapot kirajzoló objektum. A kirajzoláshoz bizonyos alapértékekkel rögzíteni kell a sarkokat, és a kezeléséhez le kell foglalni a méretének megfelelő memóriaterületet. Azután kezdjük el bővíteni különböző irányokba: készítsünk egy keretes téglalapot, amely szabadon ide-oda rakható a képernyőn, meg egy olyat, amely képes kezelni az egerünket. Ehhez persze már valamilyen felirat is szükséges, amelynek kiíratása együtt jár a szövegkoordináták feljegyzésével, és az sem árt, ha van valamilyen alapértelmezett szöveg, amellyel el lehet indítani a szövegmegjelenítést megvalósító rutinokat. A szöveget persze nemcsak nyomógombokon kell kiírni, hanem más képernyőterületeken, például címkéken is, amihez további objektumleszármazottak szükségesek.

Ahogy bonyolultabbá válik az alapobjektum funkcionalitása, úgy nő a megjelenítéséhez szükséges idő (le kell futniuk a különböző inicializáló eljárásoknak), és a használatához szükséges memória. Ennek ellenére ilyen rendszert használó program megírható a DOS bűvös (640 K-s) memóriakorlátja alá is, amit a különböző grafikus héjprogramok jól demonstrálnak.

Próbáljunk meg ezekből az objektumokból egyszerre minél többet megjeleníteni, esetleg a grafikai kinézetet is finomítva. Hamarosan eljutunk valamilyen Windows-szerű képződményhez, egyúttal pedig gépünk erőforrásainak határához. Ugyanez történik a Windows rendszerével is, amely szintén az említett módon épül fel, ezért a 16 bites Windowsok és a Windows 9x sorozat működése mögött mindig megtaláljuk a DOS-t mint tényleges operációs rendszert. A Windows több látható objektuma és jó néhány háttérbeli szálon futó funkciója magyarázattal szolgál arra, hogy a gépbe való betöltődése, a rendszer „felállása" miért olyan időigényes, és megnyitott alkalmazásai (téglalapjai) függvényében miért képes a gépünkbe helyezett memóriablokkok és a merevlemezen létrehozott virtuális memóriaterületek határáig nyújtózkodni.

A programozó is ember

Az objektumorientáltsággal kapcsolatban érdemes még néhány dologra kitérni. Az egyik, hogy egy grafikus program megírható objektumorientált programozás nélkül is, csak nehezebben. A másik, hogy az objektumorientált nyelvek használata nem az inteles, DOS-on alapuló világ sajátsága. Maga

a C++ is több platformra lett átültetve, a Linux fejlesztésének szintén ez az alapja, és az objektumorientált programozást több más nyelv is támogatja. A harmadik, hogy az objektumorientált fejlesztés nemcsak a grafikus felületek létrehozásában kap szerepet, a Java esetében például szinte minden változó objektumként létezik, amelynek az interpreter futásidőben foglal helyet, és szintén a Java futtatóprogram gondoskodik a „szemét" eltakarításáról, ha egy objektumra nincs többé szükség.

Arról sem kell megfeledkezni, hogy a Windowson kívül több más programrendszer is grafikus felületen jelenik meg. Alapértelmezésben ilyen az Apple Macintosh, vagy választhatóan grafikus felületű a Linux. Minden grafikus futtatófelület közös vonása és létrehozásának elsődleges célja a kezelés megkönnyítése vizuális eszközökkel és pixeles felbontású pozicionálással (egér). A grafikus felület az alkalmazások használatát, a felhasználó életét igyekszik kényelmesebbé tenni, és sok szempontból a programozó is csak egy ilyen felhasználó a sok közül. Máig is megvan a lehetőség arra, hogy egyszerű szövegszerkesztővel Windows felületre írjunk programokat, de ez egyre kevesebb embert tud lelkesíteni.

A kódírás szép lassan kikerült az egyéni önmegvalósítás bűvköréből. Az alkalmazásfejlesztésben éppúgy szemponttá vált a fejlesztési munka hatékonysága, mint egy gépgyárban. Ez volt az a tényező, amely nem kedvezett az olyan — a programok futásidejében és terjedelmében egyébként hatékony, de a fejlesztés szempontjából időigényes és kevésbé jól tanulható — nyelveknek, mint az Assembly. Abban az esetben sem, ha a gépfüggetlenség egyébként nem szempont. Ugyanakkor a kódolónak is könnyebbség, ha olyan környezetben végezheti a munkát, amelyben majdani programja futni fog.

A Windows esetében mindez elvezetett a windowsos fejlesztőeszközök megjelenéséhez, amelyek karakteres, DOS-os társaikhoz hasonló pályát futottak be. Eleinte alig voltak többek, mint a korábbi (karakteres) fejlesztőeszközök windowsos szövegszerkesztővel kiegészített utódai, és a további funkciókat, a fordítást és a hibakeresést is Windowson elvégző eszközök. Képesek voltak a vágólap kezelésére akár saját clipboard formátum definiálásával is, reagáltak a Windows üzenetvezérelt rendszerhívásaira, és maguk is integrálódni tudtak a rendszerbe, kapcsolatot tartva más programokkal. Vagyis a fejlesztők minden olyan funkciót beépíthettek alkalmazásukba, amelyeket a Windows programoktól elvártak. A fejlesztés során azonban manuálisan kellett megtervezni és kódolni az egyes objektumok megjelenését és viselkedését, a főablaktól a beviteli és kapcsolómezőkig, vagy a „fogd és vidd" technika megvalósításáig. De például az ikonok, dialógusablakok, menük szerkesztéséhez már külön erőforrás-szerkesztőkre volt szükség, és az így elkészült erőforrásfájlokat be kellett építeni a programba is.

„Látványorientált" uniformis

A fejlesztőeszközökkel kapcsolatos igények logikus lépcsőfoka volt, hogy a windowsos alkalmazásokban elterjedt megoldásokat és automatizmusokat a programkészítők is felhasználhassák. Elég legyen megmondani, hogy hová szeretnénk egy nyomógombot elhelyezni, és mi történjen, ha megnyomják azt. A fejlesztőeszköz vegye le a sablonos programozási munkákat a fejlesztők válláról. A megtakarított időt pedig célszerű a fejlesztésre és a hibakeresésre fordítani.

A „visual", azaz „látványorientált" fejlesztőeszközök kialakítását tulajdonképpen ez motiválta, bár a kezdeti verziók inkább csak integrált kódszerkesztők voltak. Fokozatosan kialakult azonban bizonyos szabvány az ilyen fejlesztőeszközökre. Akár a Microsoft Visual Basic és Visual C++, akár a Borland Delphi és C++Builder eszközeit tekintjük, jellegzetes építkezési módjuk, hogy az alkalmazás ablakainak kialakításához tartalmaznak egy kódszerkesztő ablakot és egy komponenspalettát, ahonnan kiválaszthatók az induláskor általában szintén megjelenő üres alkalmazásablakban (form) elhelyezni kívánt alkatrészek. A komponenseket egyszerű „fogd és vidd" technikával helyezhetjük el a form szerintünk legcélszerűbb helyén.

A későbbiekben a formban elhelyezett elemek viselkedésének programozásához a fejlesztőeszközök tartalmaznak egy további ablakot, amelyben a komponensek jellemzői, paraméterei adhatók meg. A felhasználó ténykedésére adott válaszokat, például a gomb megnyomásakor bekövetkező programlépéseket a kódszerkesztőben tovább finomíthatjuk.

A vizuális fejlesztőeszközökben már menet közben megoldható a legtöbb olyan feladat, amely korábban az erőforrás-szerkesztő kompetenciája volt. Így röptében megadhatjuk, hogy az alkalmazás milyen ikont használjon (annak megrajzolásához azonban továbbra is szükség van egy grafikus szerkesztőprogramra). Szintén menet közben van lehetőség további formok kialakítására (amelyekből elkészíthetjük az alkalmazás dialógusablakait), vagy a menük kialakítására, akár az alkalmazás főmenüjéről, akár az ún. pop-up (beugró) menükről legyen is szó. Ha pedig nagyjából készen vagyunk, akkor külön eszközsor áll rendelkezésünkre, melyen elérhetjük a fordítást, futtatást, tesztelést lehetővé tevő funkciókat.

E szerkezet általános elfogadottságát jelzi, hogy amikor újabb nyelv jelenik meg, annak támogatásához a vizuális eszközök legtöbbször hasonló felépítésűek. A Delphi és C++Builder utáni JBuilder nagy vonalakban szintén ezt a sémát követi. A vizuális fejlesztőeszközök készítése természetesen nem a fejlesztőeszközöket gyártó „nagyok" privilégiuma. Ezt mutatja a Visual Assembler esete, amely az Interneten szerveződő, szabadon hozzáférhető projekt. Támogatja a gyakori Assembly-fordítókat, a szabad forráskódúakat is beleértve. És itt visszakanyarodhatunk a nyelvválasztásnál érintett témára, hogy nemcsak saját nyelv, hanem a nyitott forráskódú rendszerek talaján állva saját fejlesztőeszköz készítésére is van lehetőségünk.

Ha azonban ennyire egységesülnek, uniformizálódnak egy adott platform fejlesztőkörnyezetei, sőt akár több platformé is, akkor merre érdemes haladni? Akinek nincs ideje az alapoktól építkezni, az bizony válassza inkább azt, amelyikben a legnagyobb gyakorlata van. A közös nevezőre hozást pedig bízza a fejlesztőeszközre, ami akár egy platformfüggetlen kód (például a Java bájtkód) létrehozását is jelentheti.

Azt azonban mindig vegyük figyelembe, hogy a komponensalapú építkezésnek ára van, amit a hardverforgalmazóknál fizetünk meg. A gyári és az általános célra széles körben hozzáférhető komponensek ugyanis nem kimondottan nekünk készültek. Az igényeinknek megfelelő képesség mellett bőségesen tartalmaznak általunk talán soha fel nem használt funkciókat, szabályozható tulajdonságdefiníciókat, változókat stb. Ezek pedig a feltétlenül szükségesnél nagyobb erőforrásigényben és időszükségletben realizálódnak, ami különösen akkor lesz feltűnő, ha a formon sokszor előfordulnak. Ilyenkor a bennük felejtett gyári hibák hatása is hatványozottan jelentkezhet, és azokat csak akkor tudjuk kijavítani, ha a forráskóddal is rendelkezünk.

Simay Endre István

A 21. század előfutára

Párhuzamos programozás

Egy problémának többféle megoldása létezhet, különösen ha nem egyetlen szempontot tartunk szem előtt. Amiből persze már az is következik, hogy időnként felbukkanhatnak olyan új szempontok is, amelyek fényében át kell értékelnünk addigi felfogásunkat. Ilyesmi történt a párhuzamos programozás megjelenésekor is.

Az 1940-es évek óta a számítástechnikát meghatározó ún. klasszikus vagy Neumann-architektúra működési képlete, hogy a központi processzor egymás után beolvassa az adatokat, egyenként elvégzi rajtuk a szükséges műveleteket, majd pedig kiírja a végeredményt a memóriába. Ez szigorúan soros egymás utáni műveletek jelent, az így dolgozó komputereket szokás SISD gépeknek is nevezni (Single Instruction Stream, Single Data Stream). Daniel Hillis, a párhuzamos programozású Connection Machine megalkotója szerint az tette szükségessé ezt a felfogást, hogy kezdetben a processzor (például a vákuumcső) viszonylag gyors volt, de drága, a memória viszont lassú és olcsó, logikus törekvésként igyekeztek tehát „kihasználni a drága vákuumcsövet, amennyire csak lehet". Ez olyan sikeres megoldásnak bizonyult, hogy a számítógép-tervezők még akkor is ragaszkodtak hozzá, amikor már technikai (és költségvetési) szempontból nem lett volna szükséges elkülöníteni egymástól a memóriát és a központi végrehajtóegységet (a processzort).

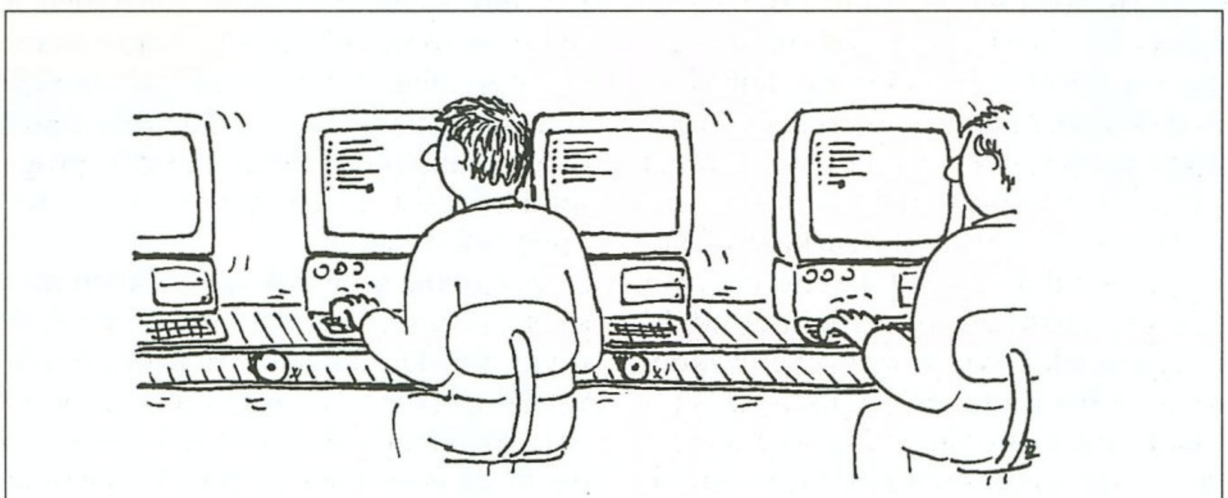
A jól bevált tervezési konvenciókhoz való ragaszkodáson kívül persze néhány más ok is közrejátszhatott a Neumann-architektúra továbbélésében. Elsősorban talán az, hogy a számítógép eredetileg csak nagy sebességű aritmetikai segédeszköznek készült, amely képes például egy lövedék pályáját kiszámítani, de lényegében a matematikusok által használt módszereket másolja le. Márpedig a matematikusok többnyire az egyik számítási eljárás kimeneti eredményét használják a másik számítás bemeneti értékeként, vagyis egyfajta soros logika szerint dolgoznak. (De a klasszikus logika is sorosan, nem pedig párhuzamosan dolgozza fel az eseményeket: ha A, akkor B; és ha B akkor C, tehát ha A, akkor C stb.) És hogy még nyilvánvalóbb legyen, miről is van szó, a kozmológus Paul Davies ehhez azt is hozzáteszi, hogy a 20. század elejének nagy matematikusa, David Hilbert mindenkit felszólított „olyan módszeres eljárás kidolgozására, amelynek révén véges számú lépésben eldönthető egy adott matematikai állítás igazsága vagy hamissága... Személyekre voltaképpen nem is volt szükség, mert az eljárás gépesíthető volt". Ilyen elméleti alapokra húzott épületben tehát kellett hozzá egy kis merészség, hogy valaki a párhuzamos programozás ötletével álljon elő.

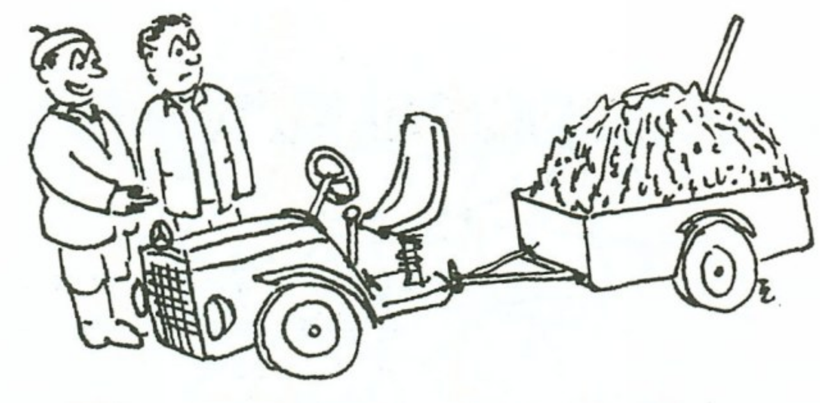
Decentralizált rendszerek

Ha a programozás nem is, de az élet szép számban produkált olyan dolgokat, amelyeket egyszerűbb nem a matematika és a matematikából kinövő számítástechnika egyetlen adatfolyamba zsúfolt (mondhatni centralizált) megközelítési módjával értelmezni. Ott vannak például a rajban repülő madarak vagy a termeszboly dolgozói: a tudósok hosszú ideig feltételezték, hogy mind az egyik, mind a másik esetben kell léteznie valamiféle irányítónak, mely eldönti, hogy merre repüljenek a madarak, vagy mit csináljanak a rovarok. A helyzet azonban az, hogy miként a közlekedési dugónak sincs központi szerve, ugyanúgy számos komplex viselkedési minta is leírható enélkül. Hogy mégis hajlunk a centralizált megközelítésre, annak feltehetően pszichológiai magyarázata van. A kísérletek szerint már a gyerekekben is erős az a feltételezés, hogy a történéseket mindig „valaki csinálja", még az esőt, a szelet, a naplementét is. Úgy tűnik hát, hogy alapértelmezésben egyszerűbb a világot saját hasonlatosságunkra elképzelni, és azt gondolni, hogy a dolgok ugyanolyan „centralizáltan" működnek, mint mi, akiknél látszólag egyetlen központi szerv, az agy dirigál.

A mesterséges intelligencia kutatása sem véletlenül választotta a centralizált megközelítést. Az 1960-as évek közepétől készített modellek célja az volt, hogy egy „egyszerű és koherens modellt alkossanak a világról", ahogyan egy helyütt a párhuzamos programozású StarLogot készítő Mitchel Resnick fogalmazott. A szakértői rendszerekben ugyanekkor egy ún. „inference engine" (kb. indukáló motor) próbált meg következtetni és döntéseket hozni a „tudásbázisban" felhalmozott, gyakran emberi szakértőktől származó információk alapján (és eközben nehéz nem észrevenni a hasonlóságot a processzor vs. memóriafelosztás között). Pedig nem ez lett volna az egyetlen lehetséges megoldás. A 80-as évek közepe óta már egyre többen állnak elő nem centralizált modellekkel.

Marvin Minsky híres könyve, a Society of Mind (1987) azt állítja, hogy az értelem egymással együttműködő és vetélkedő alrendszerek (afféle ágensek) hatására jön létre, központi irányítás nélkül. Nem sokkal később Rodney Brooks olyan robotokat konstruált, ahol egészen egyszerű és egymástól függet-





— ...meg Zsiguli sebváltó, meg Trabant kilométeróra... szóval olyan, mint egy korszerű szoftverrendszer.

len „viselkedési modulok" (mozogni, megállni, elkerülni stb.) hozzák létre a komplex — és a korábbi, centralizált modellekénél sokkal hatékonyabb — magatartást. A decentralizáltságnak megvan a maga filozófiája is: Daniel C. Dennett úgy véli, hogy amikor az értelem létrejöttéről beszélünk, akkor egyszerűen hibásan tesszük fel a kérdést, mert „nincs otthon senki", és bár úgy tűnhet nekünk, hogy létezik az öntudat meg a személyiség, erről a valóságban szó sincs. Ugyanúgy, ahogy a repülő rajnál sem valamilyen központi irányítás gondoskodik arról, hogy együtt maradjanak a madarak, hanem Craig Reynolds híres boid (azaz birdoid) modelljének három nagyon egyszerű szabálya: (1) el kell kerülni az ütközést a többiekkel, (2) velük azonos sebességgel kell haladni, (3) minél közelebb kell repülni hozzájuk. Mitchel Resnick egyenesen arról beszél, hogy eljött a decentralizáció korszaka a mesterséges intelligencia kutatásától a politikáig bezárólag mindenütt. Az amerikai Freedom House nevű szervezet arról számol be, hogy az 1990-es évek elején fordult elő a történelem folyamán először, hogy többen éltek demokráciában, mint diktatórikus rendszerben.

Connection Machine

Akadhatnak persze, akik nem értenek egyet a szélsőségesen centralizációellenes felfogással, de ők sem vitathatják, hogy a decentralizáció a számítástechnikában komoly előnyökkel járhat. Alig harminc évvel ezelőtt persze senki sem gondolt volna erre, a számítógép a központosított hatalom szinonimája volt, és legfeljebb néhány „kiválasztott" használhatta. Még 1972-ben is alig 150 000 számítógép létezett a világon — ma már viszont több száz millió. Ezzel párhuzamosan lejátszódott két „decentralizációs forradalom" is: a személyi számítógép és az Internet elterjedése. Ami az utóbbit illeti, az nem csupán decentralizált, hanem legalább 3-4 szintű redundanciával rendelkező, „elosztott" (distributed) hálózatot eredményezett, vagyis olyan hálózatot, ahol egy-egy cím legalább 3-4 különböző útvonalon érhető el, és ehhez egy Warren McCullogh nevű kutatónak köszönhetően elsősorban az emberi agy neuronhálózata szolgált mintául. A PC-k elterjedésének köszönhetően pedig a komputer már nem a „Nagy Testvér" eszközének, hanem egyre inkább környezetünk részének számít — bár belülről még mindig erősen centralizált, és a személyi számítógépben rendszerint egy (legfeljebb két) processzor végzi a munkát.

Pedig Daniel Hillis már 1985-ben úgy gondolta, hogy ideje a régi processzor-memória felosztáson túllépni, mert az kezd visszahúzó erővé válni. Az egyik kivezető utat a párhuzamos programozás, egészen pontosan az általa tervezett Connection Machine jelentheti, amelynek két alapvető sajátossága van: egyrészt sok processzort tartalmaz („I. számú követelmény: a probléma nagyságával összhangban elegendő processzornak kell rendelkezésünkre állnia"), másrészt a gép topológiája hozzáilleszthető a megoldandó probléma topológiájához („II. számú követelmény: a processzorokat szoftveresen kell összekapcsolni egymással"). Ennek megfelelően a Connection Machine-be nagyszámú olyan processzor/memória van beépítve, amelyet programozható kommunikációs hálózat köt össze. „A cellák adatfüggő mintázatokká vannak összekapcsolva, és azokat aktív adatstruktúráknak nevezzük", melyeket a Connection Machine-en kívülről egy hagyományos „host komputer" irányít. Vagyis maga a Connection Machine mintha csak ennek a gazdagépnek a memóriája lenne — azzal a nem elhanyagolható különbséggel, hogy a hagyományos megoldással ellentétben itt nem jelenik meg a Neumann-féle szűk keresztmetszet, mert a memóriacellák maguk végzik el a számításokat is. „Mivel cellák ezrei vagy akár milliói dolgoznak egyszerre a probléma megoldásán, a számítási sebesség sokkal nagyobb, mint amekkora a hagyományos gépeken lehetséges volna" — állapítja meg Hillis.

Magáról a gépről még annyit, hogy a prototípus 65 536 darab, egyenként 4096 bit nagyságú processzor/memóriacellát tartalmazott (vagyis összesen 32 milliót). A programozási nyelvről pedig először is azt, hogy a Connection Machine a Lisp egy kiterjesztett verzióját, a párhuzamos programozásra alkalmas CmLispet futtatta (ami természetesen Connection Machine Lispet jelent). Az átdolgozásra azért volt szükség, mert bár normál körülmények között „könnyű elfelejteni, hogy a hagyományos nyelvek mennyire kötődnek a hagyományos komputerek hardveréhez — még az olyan »magasszintű« programnyelvek is, amilyen a Lisp", azért aki egy párhuzamos felépítésű architektúra esetében ezt nem veszi figyelembe, az egyszerűen nem lesz képes kihasználni a gépet.

Párhuzamosság hagyományos gépeken

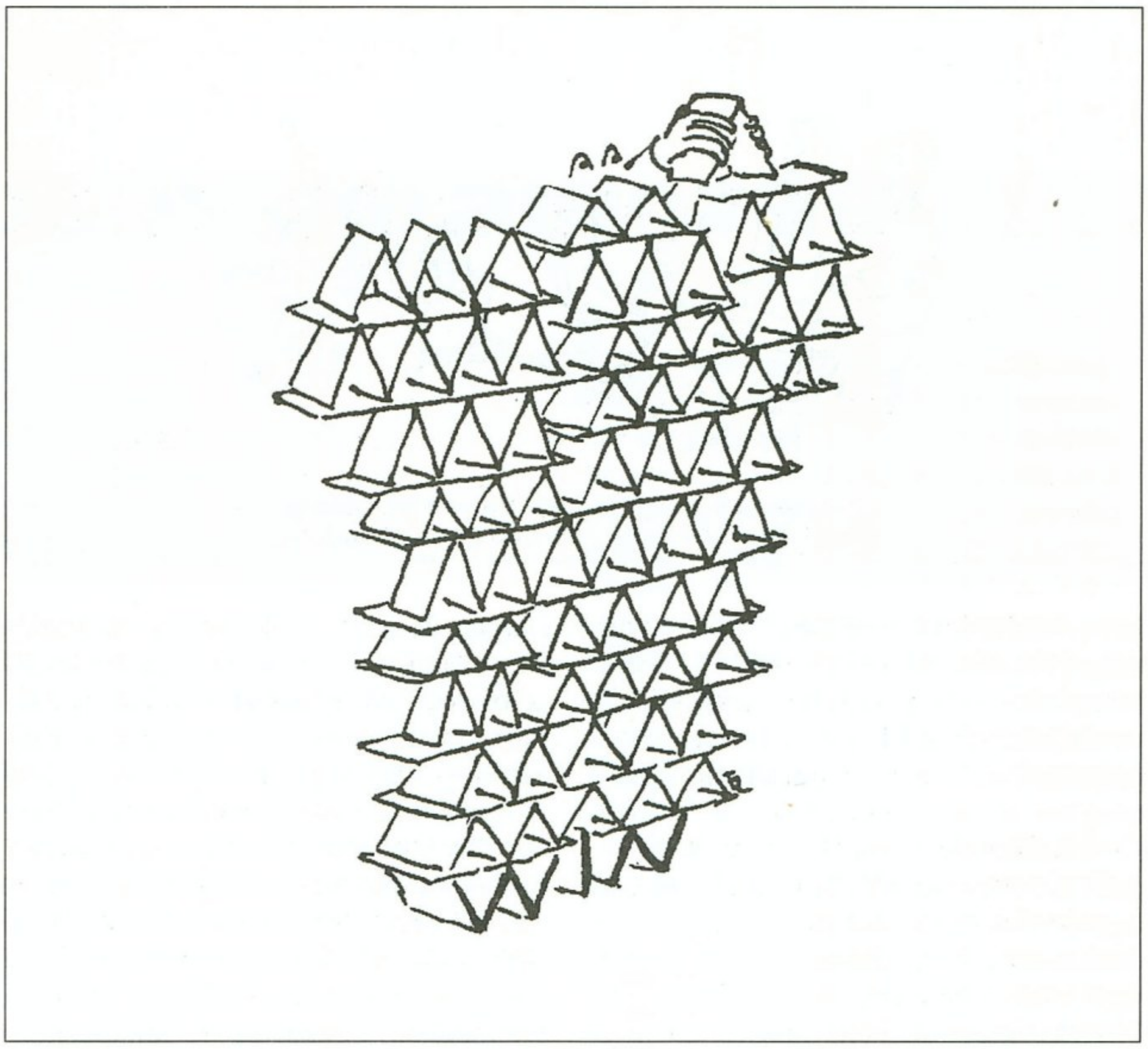
Egy hagyományos SISD architektúrán persze nyugodtan futhat párhuzamos logikájú program is, hiszen — lévén szó minden lehetséges számítógép modellezésére képes Turing-gépről — ennek semmi akadálya nincs. Sőt, olykor kimondottan előnyös is lehet, amennyiben a decentralizáltsághoz és megosztottsághoz közelebb álló feladattal van dolgunk — ahogyan az a Logo bizonyos verzióival is történt.

A Seymour Papert nevéhez fűződő Logo a 60-as évek végén tűnt fel, és elsődleges célja az volt, hogy egészen kis gyerekek számára is megtanulhatóvá, sőt vonzóvá tegye a programozást. Ezt Radia Perlman egy mechanikus, nem csupán a számítógépben, hanem a valóságban is létező teknőc irányítására használta (forward, back, left, right és hasonló, többé-kevésbé angol nyelvű parancsok segítségével), a következő lépésben pedig már mint grafikai eszközt alkalmazták. Aki a Logóval dolgozott, az a képernyőn ballagó teknőcöt képzelt maga elé, amely akkor rajzolt, ha a feje le volt engedve (és így a szájában tartott ceruza a papírhoz ért).

1985-ben azután Resnick, Papert és egy harmadik számítógépes, Steve Ocko arra gondolt, hogy érdemes volna a Logót a LEGO-val kombinálni, vagyis lehetővé tenni, hogy a LEGO-elemek-

ből összeállított gépeket az erre a célra kiválóan alkalmas Logóval vezéreljék, és hamarosan létre is jött a LEGO/Logo rendszer, amelyet azóta is számos amerikai iskolában használnak (kár, hogy Magyarországon nem). Eközben persze be kellett vezetni a fizikai valósággal kapcsolatban lévő parancsokat, amilyen a „go-until-bump" (menj, amíg neki nem ütközöl valaminek) vagy a „sensor" (amivel az érzékelő állapotát lehetett lekérdezni). Hamarosan az is kiderült, hogy a hagyományos, soros felfogású Logo segítségével gyakran jobb esetben is csak bonyolultan és nehézkesen lehetett megoldani azokat a problémákat, amelyek párhuzamos programozást használva nem jelentettek volna gondot, például egy robot két kezét egyszerre mozgatni.

A kiutat a MultiLogo jelentette, az ún. ágensek bevezetésével. A programozó ettől kezdve egyszerre több processzt is futtathatott, mivel minden ágens a MultiLogónak a többi, éppen futó és a MultiLogótól függetlenül működő verziójaként viselkedett. Egyszerre „számos ágenst használva, a felhasználó számos folyamatot tarthat ellenőrzése alatt" — írja Teknőcök, termeszek és közlekedési dugók c. könyvében Resnick, aki a már kereskedelmi változatban (MicroWorlds néven) létező MultiLogótól két ponton is alapvetően eltérő StarLogót írta meg, és amiből azután az e havi CD-mellékleten is megtalálható, és Windows alatt működő StarLogoT-t fejlesztették ki (az eredeti kizárólag Unix és Macintosh alatt futott, ez viszont egy windowsos alfa-változat).



A StarLogo két alapvető vonásában is különbözik a MultiLogótól, mert fejlesztése közben teljesen más szempontokat tartottak szem előtt. A MultiLogo viszonylag kevés, de párhuzamosan futó processz kezelésére való, a StarLogo viszont egyszerre több száz, ha ugyan nem több ezer párhuzamosan futó folyamatot (amilyen a hangyák ténykedése a bolyban) képes kézben tartani. Másfelől míg a MultiLogo különböző ágensei gyakran alapvetően különböző tevékenységeket szabályoznak (például az érzékelők lekérdezését, miközben a kéz megfog valamit), addig a StarLogo leginkább arra alkalmas, hogy nagyszámú, hasonló folyamatot „stabilan párhuzamos" (massively parallel) módon kezeljen. Hasonlatként ismét csak a boly hangyáit említhetjük.

És ha már itt tartunk, említsük meg a StarLogo néhány további sajátosságát is. Miközben a hagyományos Logo elsősorban a geometria felfedezésére szolgáló „rajzolónyelv", ahol a teknőc a szájában tartott ceruzával kizárólag a programozótól érkező utasításokat követi, és vakon mozog a képernyőn (a „nekiütközni" fogalma például nincs is értelmezve, hiszen nem tud minek nekimenni), aközben a StarLogo teremtményei képesek arra, hogy például más teknőcök jelenlétét érzékeljék. Sőt, rendelkeznek egy „beépített tulajdonsággal", melynek révén képesek „szagnyomokat" követni. A StarLogo másik alapvető vonása ugyanis az interaktív környezet. A világ kisméretű négyzetekre, ún. „patch"-ekre (szó szerint „foltokra") van felosztva, és ezek — attól eltekintve, hogy nem képesek mozogni — a teknőcökhöz hasonlóan nagyon is sokféle állapotba kerülhetnek, és nagyon is sokféle információt tárolhatnak. „Leginkább olyanok, mint egy sejtautomata cellái" — állapítja meg Resnick. Ez pedig nem csupán azt teszi lehetővé, hogy a patchek információkat tároljanak például a környezetben található élelmiszer mennyiségéről, hanem azt is, hogy a teknőcök „szagnyomokat" húzzanak maguk után, melyeket társaik érzékelőik segítségével fel tudnak fogni, és így kialakulhat egy (például a termeszbolyokra is jellemző) nem közvetlen, nem egyedtől egyedig történő, hanem a környezetet közvetítőként felhasználó kommunikációs forma. És hogy a lehetőségek skálája teljesebb legyen, a teknőc-patch, a patch-teknőc és a teknőc-teknőc mellett a patch-patch kommunikáció is megengedett. De persze nem az egyedek, hanem az egyedek viselkedéséből kirajzolódó, „kóloniaszintű" magatartáson van a hangsúly, és ez teszi kiválóan alkalmassá a StarLogót komplex, sok azonos vagy egymáshoz hasonló elemet tartalmazó rendszerek paralel modellezésére.

Nem csoda hát, hogy számos hozzáértő szerint a StarLogo már a 21. századi programnyelvek előfutára, hiszen könnyen és gyorsan tud megoldani párhuzamos megközelítést igénylő feladatokat. D. E. Knuth mondja egyik könyvében, hogy „a számítógépek és a matematika kapcsolata sokkal mélyebb, mint az a hagyományosnak mondható kapcsolatok alapján sejthető". James Bailey a mesterséges intelligencia történetéről szóló könyvében egyenesen azt állítja, hogy Newton fizikai alapműve, a mondhatni tökéletesen soros logikára épülő Principia Mathematica vált az ENIAC és az utána következő elektronikus komputerek... tervezési specifikációjává".

Ma már azonban azt is tudjuk, hogy a hagyományos, az emberi logika működéséhez idomuló megközelítési mód nem mindig idomul tökéletesen a valósághoz.

Galántai Zoltán*

* A szerző a cikket az MTA Bolyai-ösztöndíjasaként írta, a StarLogóval kapcsolatos anyagok összeállításában Mákos András is közreműködött.

A Basic, a Minix és a Linux

A programozás esztétikája

A programozás története értelmezhető annak a kérdésnek a tanulmányozásaként is, hogy az egyszerűségre és a könnyű áttekinthetőségre való törekvés meddig szolgálja a használhatóságot és a sokoldalúságot. Szinte teljesen biztos ugyanis, hogy nem lehet egyszerre minden feltételt teljesíteni.

Egy Charles Kay Ogden nevű nyelvész 1930-ban azt javasolta, hogy a világ népei válasszák egyetemes közvetítő nyelvnek az angolt, és hogy ne kelljen sem a nyelvtannal, sem a nagy szókinccsel bíbelődni, meg is alkotta az ún. Basic Englisht, amely alig 850 szóra épült. A név egyúttal szójátékot is rejtett magában, a Basic ugyanis ez esetben az eredeti angol „alapvető" értelem mellett annak a rövidítése is volt, hogy „British American Scientific International Commercial" (vagyis: brit amerikai tudományos nemzetközi kereskedelmi).

Jóllehet Ogdent olyan nagyságok is támogatták, mint Churchill vagy Roosevelt, a javasolt nyelv soha nem tudott elterjedni, mert erőssége egyben a gyengesége is volt: ahhoz, hogy nagyon kevés kifejezéssel is nyelvi formába lehessen önteni a mondanivalót, roppant bonyolult kifejezéseket kellett megalkotni. Az embernek rögtön a Basic számítógépes nyelv jut az eszébe, ahol mintegy ötven kifejezés segítségével állítólag mindent be lehet programozni, de semmit sem olyan egyszerűen, hogy megérje használni (vagy hogy az a gyakorlatban jól kivitelezhető legyen).

A szabadszoftver-koncepció guruja, Eric S. Raymond által szerkesztett és sok számítástechnikai bölcsességet magában foglaló Jargon File a Basic programozási nyelvről egyenesen azt állítja, hogy az a „reménybeli hackerek jóvátehetetlen agykárosodásának okozója". Idézi Edsger W. Dijkstra számítógépes szakértőt is, aki kijelenti (Válogatott írások a számítástechnikáról: egy személyes nézőpont c. művében), hogy „Gyakorlatilag lehetetlen megtanítani a helyes programozási stílusra azokat a diákokat, akik korábban a Basic romboló hatásának voltak kitéve: mint potenciális programozók olyan mértékben sérülnek, hogy gyógyulásukra semmi remény". (A programozók szerint egyébként ugyanez a baj az eredetileg szintén oktatási célokra készült Pascallal is.)

Ha egy programnyelv alkalmas rá, hogy azon komolyabb előkészület nélkül gyakorlatilag bárki megírjon egy 10-20 soros programocskát, akkor ebből majdnem egyenesen következik, hogy komolyabb feladatokra nem is lehet majd felhasználni. Tulajdonképpen nem nehéz érveket találni a Basic ellen. Ott van például a meghatározott helyre való ugrást lehetővé tevő GOTO, amely látszólag ugyan megkönnyíti a munkát, ennek azonban az a következménye, hogy mindenféle logika figyelmen kívül hagyásával, teljesen önkényesen járhatunk el, és a program pillanatokon belül nem csupán a kívülálló számára, hanem számunkra is áttekinthetetlenül kusza lesz. Mivel bárhova odaugorhatunk, a korlátlan szabadság lehetősége szükségképpen lerombol minden rendet. Márpedig ez ebben a műfajban (is) megengedhetetlen.

Ami működik, az csodálatos

A technika esztétikájáról könyvet író David Gelernter professzor úgy véli, hogy „A szépség nélkülözhetetlenül fontos a számítógépes szakembereknek, mindenekelőtt azért, mert a virtuális gépeket örökösen az a veszély fenyegeti, hogy elnyeli őket önnön komplexitásuk." A fizikailag is létező számítógépeket (vagyis a hardvert) a fizikai valóság korlátozza, mert ha túlméretezzük, akkor beszakad alatta a padló, vagy a túlterheléstől leég a hálózat. Gondoljunk azonban a Windows 2000 nem kevesebb, mint 40 millió kódsornyi programjára! Fizikailag semmi akadálya nincs egy ilyen monstrum megalkotásának, miközben Gelernter szerint egy modern autó bonyolultsága mindössze „egy 30 000 soros programmal egyenértékű". Eközben a felmérések szerint a kereskedelmi szoftverek átlagosan 50%-kal hosszabbak a szükségesnél (vagy az elfogadhatónál), és minden negyedik fejlesztés félbemarad, aminek megvan a maga jó oka.

Egy Windows 2000 méretű, monolitikusan felépülő szoftver — de egy annál lényegesen kisebb is — nem csupán áttekinthetetlen, hanem menthetetlenül működésképtelen is. Ellenpontként a modulárisan, egymástól független elemekből összeálló Linuxot szokták emlegetni, mint a „szépség diadalát a pusztán pénzszerzésért írott szoftverek felett", és Gerald M. Weinberg, a számítógép-programozás pszichológiájának szakértője úgy gondolja, hogy „a program esztétikai és gyakorlati értéke közötti összefüggés nem véletlen: minél inkább tetszik a szemnek és az észnek, annál valószínűbb, hogy jó a program. Vagy hogy költőien fogalmazzunk: »A szép igaz, az igaz szép«". Amiből következik, hogy a Windows típusú megoldások nem ilyenek: a látványosan könnyen kezelhetőnek tűnő felhasználói felületért cserébe egy működési ellentmondásokkal terhes rendszert kapunk, annak szükségszerű és kiküszöbölhetetlen hátrányaival együtt. Másfelől viszont bizonyos szempontból a Linux is messze van a tökéletességtől.

Az ideális rendszer

Milyennek is kellene lennie az ideális operációs rendszernek? Erre egyáltalán nem egyszerű válaszolni. Mert szavazhatunk ugyan a „szépségre" a monolitikus konstrukció helyett, ám könnyen előfordulhat, hogy a tökéletes programnak egymást kizáró feltételeket kellene teljesítenie. Ismét csak Weinberg szerint: „Ugyanazok a vezetők, akik hatékonyságért rimánkodnak, a hajukat tépik akkor, amikor értesülnek a módosítások költségeiről. Azok a vezetők viszont, akik általános és könnyen módosítható programokat akarnak, panaszkodnak, amikor megtudják, hogy milyen lassúak és időigényesek az elkészült programok."

A Macintosh annak idején valóban könnyű és intuitív operációs rendszeréről volt nevezetes, mára azonban annyi mindent zsúfoltak bele, hogy már komoly szakértelem kell hozzá. Egy operációs rendszertől képtelenség elvárni, hogy egyrészt legyen könnyen kezelhető és egyszerű, másrészt legyen alkalmas



— Így ugye sokkal jobb lesz a programozói környezet!?

bonyolult beállításokra és összetett szolgáltatások nyújtására. A kérdés ilyen esetekben mindig az, hogy melyik szempont fontosabb számukra. Erről folytatott érdekes vitát 1992-ben Andrew S. Tanenbaum (aki többek között a Minix és az Amoeba operációs rendszereket tervezte, vagyis nem tévesztendő össze a két „n”-betűs Andrew Tannenbaummal) és Linus Torvalds (akinek a nevéhez a Linux operációs rendszer kerneljének elkészítése fűződik).

Ma, 1999-ben úgy tűnhet, hogy nem kérdéses, melyikük sikeresebb. A Linux rövidesen vezető operációs rendszerré nőheti ki magát, a Minix pedig nem igazán tudott előrelépni, megmaradt az a kis rendszerigényű operációs rendszer, amely „felhasználói szempontból kompatibilis, felépítésében viszont különbözik a Unixtól”, és mivel „egy évtizeddel a Unix után készült, sokkal strukturáltabb... A másik különbség, hogy a Unixot hatékonyra, a Minixet áttekinthetőre tervezték... [ám] egy IBM PC-n futó Minix felhasználója Unix-felhasználónak érezheti magát” — írja Tanenbaum.

A Minix megjelenése után nem sokkal már 40 000 fős newsgroup foglalkozott vele, és rengetegen akartak új szolgáltatásokat hozzátenni a meglévő operációs rendszerhez, hogy ezzel „nagyobbá, jobbá (de nagyobbá biztosan) tegyék azt”. A Minix csupán azért volt képes megőrizni előnyös tulajdonságait (kis méretét, világos és áttekinthető szerkezetét), mert „az operációs rendszer szerzője éveken keresztül sikeresen elhárította ezeket a támadásokat”. Azután „jött egy finn egyetemi hallgató, Linus Torvalds, aki elhatározta, hogy ír egy Minix változatot bő szolgáltatásokkal, de ez már egy éles rendszer lesz, és nem oktatási segédeszköz”.

Torvalds azonban végrehajtott néhány olyan változtatást is, melyek miatt Tanenbaum úgy ítélte meg, hogy koncepcióját tekintve a Linux már létrejöttekor elavult. A legtöbb régi operációs rendszer (például a Unix, az MS-DOS, a VMS, az OS/360 vagy a Multics) úgynevezett monolitikus kernellel dolgozik: az egész operációs rendszer „egyetlen kernel módban futó a.out fájl”. A '90-es években viszont azok a mikrokerneleken alapuló rendszerek számítanak korszerűnek, ahol „az OS nagy része különálló processzként fut, leginkább a kernelen kívül”, és a kernelnek az a feladata, hogy fenntartsa a processzek közötti kommunikációt, kezelje a megszakításokat stb. Ilyen például az Amoeba, a Chorus vagy a Mach — és természetesen ilyen a Minix is, ahol a fájlrendszer és a memóriakezelés a kernelen kívül futó két processz, és az I/O driverek szintén külön processzek. A Linux azonban egyáltalán nem ilyen, mert monolitikus kernelre épül, és Tanenbaum szerint ez olyan mérvű visszalépést jelent az 1970-es évek felfogásához, mintha „egy működő C programot átírnánk Basicbe”.

Ráadásul míg a Minixet az Intelről átírták 680x0-as processzorokra (Atari, Amiga, Macintosh), a Sparcra és NS32016-ra, addig a Linux akkoriban (1992-ben) még egyedül 80x86-os gépeken futott, és úgy látszott, hogy ez a helyzet nem is fog megváltozni. Azaz a Linux Tanenbaum szerint mintha megtestesítette volna az operációs rendszerek legfőbb hibáit.

Torvalds kiállt amellett, hogy a Linux jobb, de emellett meglehetősen durva és személyeskedő stílusban válaszolt (például hogy a PC Minix jó kódjainak nagy részét a jelek szerint Bruce Evans írta, nem pedig Tanenbaum stb.), de a teljes szóváltást nem érdemes bemutatni. A szereplőkkel kapcsolatban pedig helyesebb arra az álláspontra helyezkedni, hogy az emberek tulajdonságainak megítélését nem érdemes összekeverni produktumuk megítélésével.

Nem azonos súlycsoport

Már csak annak a kérdésnek a megválaszolása van hátra, hogy ha a számítógépes esztétika elvei szerint Tanenbaumnak volt igaza (elvégre a mikrokernel valóban „szebb és elegánsabb” a makrokernelnél), akkor miért a Linux maradt mégis felül. Nyilvánvaló, hogy a felhasználók értékítélete nem feltétlenül esik egybe a programozókéval (ami elég látványosan bebizonyosodott a Windows esetében), és ez a körülmény fokozottan érvényesül a jövőben, amikor a számítástechnika tényleg az élet minden területén megjelenik, és ennek megfelelően tényleg mindenki felhasználóvá válik. Tanenbaum olyan értékekre hivatkozik, amelyeket az átlagfelhasználó nem láthat, mert őt csak a felszín, a működés külső felülete érdekli. És nem is lehet elvárni tőle, hogy szakmai kérdésekkel bíbelődjön.

Az megint más kérdés, hogy egy esztétikai szempontból elhibázott operációs rendszer is eljuthat arra a szintre, amikor már gyakorlatilag kezelhetetlenné válik, és az átlagfelhasználó is nemet mond rá. Ám soha nem az fogja érdekelni, hogy mikrokernel vagy makrokernel működik, hogy moduláris vagy monolitikus programozás hozta-e létre az operációs rendszert... Végső soron tehát a Tanenbaum által felvetett problémák fontosak ugyan, de az operációs rendszerek íróin és az adott felületre alkalmazásokat készítő fejlesztőkön kívül keveseket érdekel igazán.

Vitathatatlan, hogy a Linux ma sokkal népszerűbb a Minixnél, és hogy a Minix áll közelebb az „ideális operációs rendszerhez”, ezért alkalmasabb az operációs rendszerek tanítására. Amennyiben ezeket a szempontokat tartjuk fontosabbnak, akkor a Minix a maga kategóriájában verhetetlen. De nem azonos kategóriákról lévén szó, értelmetlen úgy tenni fel a kérdést, hogy melyik ér többet: egy sikeres „szerveroperációs” rendszer, vagy egy jól és könnyen kezelhető „mintaoperációs” rendszer. A számítástechnikában éppen az a szép, hogy a Windows elterpeszkedése ellenére még maradt hely benne a különbözőségeknek.

Galántai Zoltán

Fejlesztőkörnyezetek a nyelvekhez

Ez az IDE nem az az IDE...

A programozási nyelvekkel végzett munkát megkönnyítik a fejlesztési segédeszközök. Használatukkal egyszerűbb a kód megírása, és futtatható kódra történő fordítása, a hibák megkeresése és kijavítása. A fejlesztőeszközök mindezt integráltan, egységes keretben teszik lehetővé a hagyományos karakteres, és az újabban elterjedt grafikus platformokon egyaránt. Az IDE rövidítés az integrált fejlesztési környezetre utal: Integrated Development Environment. (Kár, hogy ez a három kezdőbetű megegyezik egy gyakran használt hardveres rövidítéssel is!)

A programozók nagyon sokféle fejlesztőkörnyezetben dolgoznak, amit jól tükröz az Internet kínálata. Legaktívabbak azonban a nagy szoftvercentrumok (Borland, IBM, Microsoft stb.), amelyek rendre elkészítik az általuk éppen támogatott programnyelvjárásokhoz a fejlesztőkörnyezeteket, és szabadon kipróbálhatóvá teszik azokat, hiszen így sokkal nagyobb az esélye annak, hogy a munkahelyi beszerzések eldöntésénél a munkatársak által már tesztelt (és jól vizsgázott!) termékek kerülnek előnyösebb pozícióba. Ez a kooperációs készség tette lehetővé, hogy az Új Alaplap CD-mellékletén már korábban is közreadtunk több ilyen programot, elég utalni a Visual Basic vagy a JBuilder próbaverziójára, vagy legutóbb az Oberont támogató eszközökre.

Mostani lapszámunkban bővebben is olvashatnak a Borland C++ Builder eszközének legújabb verziójáról, a CD-mellékleten pedig a Javát támogató eszközparkról. Ugyancsak a CD-n található a korábbi sorozat folytatásaként a Sybase PowerJava tanulóverziója, illetve az IBM VisualAge for Java 2.0 OS/2-re írt próbaverziója. A Borland C++ Builder 4.0-t a regisztrálás körülményessége miatt nem raktuk fel a CD-re, de a cég magyarországi képviseletétől az is beszerezhető.

A nagy cégek által fejlesztőeszközökkel kevésbé támogatott nyelvek közül álljon itt még néhány, a teljesség igénye nélkül:

Assembly

A gépi kód hatékonysága továbbra is nagy kihívás a programozóknak és a nyelvfejlesztőknek. Néhány assembler szabadon is hozzáférhető, a Netwide Assemblerhez például fejlesztőeszközt fejleszt a netes közösség. Aki pedig grafikus alapon szeretné használni valamelyik assemblert, annak figyelmébe ajánljuk a Visual Assembler projektet, amely szintén a világhálón érhető el. Célja a gyors alkalmazásfejlesztés (RAD) megvalósítása, részben a C++ Builder segítségével.

C – –

Lapunkban már írtunk erről az érdekes kezdeményezésről, amelyet Peter Cellik (Sphinx) indított útjára a C szintaktikán alapuló Assembly-szerű programozás megvalósítása érdekében. Fejlesztőeszköz szintén van hozzá.

Fortran

A nyelv a nagygépes korszakban egyike volt a legelterjedtebbeknek, de PC-s implementációkban sincs hiány, beleértve a különböző műhelyekben készülő fejlesztőeszközöket, fordítókat. CD-mellékletünkön több public domain jellegű Fortran eszközt találhatnak. A legutóbbi számban említett PGI (Portland Group, Inc.) fordítót és eszközcsomagot annak 100 MB feletti terjedelme miatt nem raktuk fel, de a 15 napig kipróbálható csomagot a téma iránt érdeklődők a Weben keresztül, a fejlesztő honlapján elérhetik: http://www.pgroup.com/pgi_home.html.

Logo

Bár sokan nem tekintik komoly programnyelvnek, a Logo valószínűleg az egyik legjobb tanulónyelv. A több nyelvben is meghonosított és közismert teknőcgrafika lehetőséget ad grafikai programok készítésére. A Logo tanulását támogató fejlesztőeszközök közül CD-mellékletünkön is megtalálható néhány, mint például a 16 bites Windowshoz készült Microsoft Windows Logo és a Comenius Logo részben magyarított demója.

Pascal

A DOS/Windows platformokon fejlesztők bizonyára a Borland implementációjával találkoznak. Pedig elérhető szabadon terjesztett és több más platformra átvihető változata is. CD-mellékletünkre például feltettük a Free Pascalt.

Perl

Fejlesztőeszközökben bizony nem bővelkedik. A grafikus felületen futó PerBuildernek (SolutionSoft) azonban van 14 napig használható próbaverziója is. Ehhez magát a 32 bites Windowshoz készült Perl implementációt külön kell telepíteni. A CD-n a SolutionSoft javaslata alapján megtalálható a Pw32i316.exe fájl, de feltettük az elérhető legújabb Perl csomagot is.

Python

A Web világában terjedő szkriptnyelv, melynek javás csatlakozómodulja a JPython. A nyelvvel kapcsolatos legfrissebb információk a http://www.python.org/ címen érhetők el, a különböző platformokra szánt csomagokkal együtt. Ezek között a fejlesztést segítő windowsos implementáció is található, bár fejlesztőkörnyezete egyelőre még nem ad olyan kiszolgálást, mint a többi nyelvé.

TCL/TK

A Perlhöz hasonlóan elterjedt szkriptnyelv, amelynek használatát a különböző platformokra kidolgozott implementációk és eszközök segítik. Ezek rendelkezésre állnak Windowsra, Linuxra és más Unixokra is (http://www.scriptics.com/software/).

Ha valaki a Világhálón a keresőket rászabadítja a programnyelvekre, nagyon sok lelőhelyet talál. A Yahoo tematikus oldalán a felsorolás például ki sem fér egyetlen képernyőre. (http://dir.yahoo.com/Computers_and_Internet/Programming_Languages/).

Simay Endre István

Basictől Basicig

Egységes felület

Az 70-es évek közepén a 256 bájt memóriával rendelkező Altair (Micro Instrumentation and Telementry Systems — MITS) jelentette a mikroszámítógépet. Erre kezdte meg a Thomas Kurtz és John Kemeny által 1963-ban kifejlesztett Basic (Beginners All-purpose Symbolic Instruction Code) futtatójának kifejlesztését két ifjú programozó, Bill Gates és Paul Allen. Az 1975-ben megalapított Microsoft kezdetben maga mögött tudta a kor legnagyobb számítógépgyártójának, az IBM-nek a támogatását, és ennek révén jutott el (előzőleg egy „félkész" terméket beszerezve) a személyi számítógépek legelterjedtebb operációs rendszerének, a DOS-nak első verziójához.

A Basic nyelv támogatása jelentette (és jelenti) a programozási lehetőséget az Intel szabványú platformra épült, IBM szabványú gépekre írt DOS operációs rendszer esetében is: 1987 előtt a Basica, illetve a GW-Basic, majd ezt követően a QBasic. Ez utóbbi lett az a Basic implementáció, amely megjelenését követően végig kiszolgálta az MS-DOS korszakot, egészen a csak parancssoros, karakteralapú operációs rendszerek utolsó tagjáig, az MS-DOS 6.22-es verzióig, de megtalálható még a Windows NT 4.0 rendszerben is.

A QBasic rendszer, mely a DOS telepítése után azonnal a számítógéphasználók rendelkezésére áll, szinte minden olyan alapfunkciót biztosít, amely integrált környezetű programfejlesztést tesz lehetővé. Integrált szövegszerkesztője révén folyamatos külalakigazítás is zajlik a Basic programsorok beírásakor. A beírt program egyetlen gombnyomással, a keretrendszerből való kilépés nélkül elindítható, és az indítást követően folyamatos a szintaxis- és hibafigyelés. Ebben a Basic alapú fejlesztőkörnyezetben már rendelkezésre állnak azok a nyelvi bővítések is, amelyek a fájlműveleteket megkönnyítik, illetve az önálló eljárások, függvények definiálásának lehetőségét megvalósítják.

Azonban a Microsoft az Intel alapú gépekre készült operációs rendszerek karakteres korszakában sem csak ezt az egy fejlesztőeszközt kínálta a programozóknak, hanem több programozási nyelvet is támogatott. Ezek egy részéhez a programozási eszközök fejlesztését időközben befejezték. A Microsoftnak volt Pascal, Cobol és Fortran fejlesztőkészlete is. A Cobolt és a Fortrant a Digital Equipment vette meg a 90-es években. Hosszabb ideje nincs újabb verziója az Assembly nyelvű fejlesztést lehetővé tevő MASM fejlesztőeszköznek sem.

A gépi kódú programozás szinte minden esetben hatékonyabb programkódot eredményez, mint a legtöbb magas szintű programnyelv használata. A programfejlesztők számára ezért lényeges, hogy az Assembly nyelvhez is legyen integrált fejlesztőeszköz. A MASM ezt nyújtotta, és utolsó (6.x) verziója már nemcsak a karakteres DOS-os, hanem a grafikus windowsos felületre végzett fejlesztést is támogatta, bár maga a fejlesztőkörnyezet még karakteres rendszerű volt.

Hasonló kettősség jellemezte a C nyelvet támogató eszközt, a Microsoft C-t. Ez lehetővé tette a szabványos C nyelvű programozást éppúgy, mint objektumorientált C++ nyelven végzett munkát. Természetesen mindkét nyelven a legkorábbi időktől fogva lehetett Windows programokat írni, sőt sokáig a C nyelv volt az elsődleges programfejlesztési lehetőség a Windows programok készítéséhez. Ez bonyolulttá tette a Windows programok készítését, de a Visual Basic megjelenésével a C programozóktól megkövetelt aprólékos munkát már a fejlesztőeszköz végezte el. A C/C++ fejlesztőkörnyezetek legutóbbi verziói közelítenek ehhez a szinthez, például nem kell minden esetben újraindítani a tesztelt programot, ha a forráskódon változtattunk a futás felfüggesztése alatt. Erre a karakteres QBasic fejlesztőkörnyezet használatakor is lehetőség van. A DOS programok írásának bonyodalmai azonban szinte nem is mérhetők össze a Windows alá végzett fejlesztés buktatóival, ami nagyban hozzájárult az új fejlesztőkörnyezetek kialakulásához.

A Windows megjelenésével és elterjedésével a programozók és az őket kiszolgáló fejlesztőeszközök készítői nagy kihívással kerültek szembe. A programoknak és a fejlesztőeszköznek is alkalmazkodniuk kellett a Windows üzenetvezérelt környezetéhez, olyan felületet kialakítva, hogy a program egyes eljárásait maga a Windows rendszer tudja meghívni, míg más esetekben az elkészült program legyen képes meghívni a Windows egyes belső függvényeit. Ez utóbbihoz a Windows API (Application Programming Interface) biztosítja az átjárást.

A Windows felület ráadásul olyan igényt is szült, hogy az egyszerűbb programok megírására, ötleteik megvalósítására azok is képesek legyenek, akik nem rendelkeznek a C/C++ nyelv és a Windows API függvényeinek mélyreható ismeretével. A Windows programok fejlesztését azonban nem lehet „megúszni" az API függvények és a Windows üzeneteinek kezelése nélkül. Ezt az ellentmondást oldotta fel a Microsoft a Visual Basic környezet kifejlesztésével. Ebben a fejlesztőkörnyezetben a programíró arra is lehetőséget kap, hogy néhány sor beírásával elkészítse saját működő „programját". Közben a Windows alaprendszer bonyolultságát lehetőség szerint persze elfedik előle.

Az eredmény egy olyan interpreteres megoldás lett, melyben a programozó tevékenységének eredménye egy rövid programfájl. Ez az .EXE állomány az interpreter számára előfordított kódot tartalmazza. Magát az interpretert (amely kezelte a rendszerhívásokat) egy önálló, a verziószám növekedésével egyre nagyobb méretű fájl, egy .DLL (dinamikusan kapcsolható könyvtár) tartalmazta. Ilyen futtató könyvtárak (mint a 16 bites környezetben például a VBRUN*.DLL állományok) tartalmazták a vizuális elemek többségének definícióit is.

Még a 16 bites Windows korszakban a Microsoft megkezdte a kifejezetten adatbáziskezelésre szánt fejlesztőeszközök készítését is. Ennek alapját a 90-es évek elején beszerzett FoxPro fektette le. A FoxPro azóta is a Microsoft xBase alapú adatbáziskezelője és fejlesztői környezete. (Lásd az ehhez kapcsolódó cikket a 19. oldalon.)

Mire azonban a 32 bites Windowsok korszaka eljött, a Microsoft folyamatos profiltisztítást és átalakítást hajtott végre. A karakteres környezetek fejlesztőeszközeinek fejlesztését és támogatását leállították. Napjainkra a Microsoft a 32 bites Windows környezetben dolgozó programozóknak egységes környezetet biztosító keretet alakított ki. Ez Microsoft Visual Studio néven kerül forgalomba, és jelenleg a 6.0-s verziónál tart. A már említett FoxPrón kívül megtalálható benne a hagyományos nyelvek közül a C++ támogatása a Visual C++ környezettel, és a Visual Basic legújabb verziója is. Ez utóbbiban az 5.0-stól kezdve lehetőség van valódi bináris kódot tartalmazó, és így gyorsabb .EXE fájlok készítésére, de a külső .DLL állományra a vizuális komponensek raktáraként továbbra is szükség van. A Visual Studio csomagban az említett „hagyományos" programnyelveken fejlesztőknek készült eszközökön kívül megtaláljuk a Java-fejlesztőknek szánt Visual J++ fejlesztőeszközt is.

Ezek mindegyike lehetőséget teremt a 32 bites Windowson futó programok fejlesztésére, felhasználva mind a keretrendszerrel telepített, mind a más forrásból származó külső komponenseket. Így az említett Visual Studio csomag esetében (hasonlóan a többi fejlett szolgáltatáskört tartalmazó fejlesztőcsomaghoz) fokozatosan csökken a fejlesztéshez ténylegesen használt nyelv kiválasztásának jelentősége. Az egyre nagyobb jelentőségű teammunkában a fejlesztőcsoport tagjai azon a nyelven programozhatnak, amelynek használatában járatosabbak, vagy amely a megvalósítandó cél eléréséhez célszerűnek látszik.

A csoportmunka támogatása oda vezet, hogy az egyre integráltabb programozási környezetek kiválasztásakor nem maga a programnyelv, hanem a fejlesztőkörnyezethez adott szolgáltatás válik meghatározó szemponttá. A Visual Studio csomagban ilyen szolgáltatás például a SourceSafe, mely biztosítja a különböző nyelveken végzett fejlesztések forráskódjának verziókövetését. Nyilvántartható vele a forráskódon dolgozó személy, és az általa végrehajtott változtatások tartalma, tehát minden hiba elkövetője azonosítható, ha szükséges. A SourceSafe azonban nemcsak forráskódok, hanem más szöveges dokumentumok, például HTML lapok követését is megoldja.

A forráskódokkal ellentétben kész, lefordított programegységeknek, objektumoknak a közös fejlesztésben való felhasználását támogatja a Visual Component Manager. Ez az eszköz projektszinten tartja nyilván a közös raktárban (Repository) tárolt komponenseket. Ahhoz, hogy ezek kezelése széles körben megoldható legyen, külön szabvány készült a Repositoryhoz, így más fejlesztők ezt támogató komponensei szintén elhelyezhetők benne, függetlenül a komponensfejlesztéshez használt programozási nyelvtől.

A különböző nyelvek közös, egy adott platformra specifikált fejlesztőkörnyezetbe illesztése azonban felvetett néhány más kérdést is. Olyan nyelvek esetében, amelyek kialakulásának alapja a platformfüggetlenség megteremtése volt, vajon mennyire lehet alkalmazni az új eszközöket. Ilyen a C/C++ nyelv is, amelynek gyökerei a nagygépes korszakba nyúlnak vissza. A Visual C++ környezet fordítója a Microsoft tájékoztatása szerint alkalmas a szabványos ANSI C alapú, illetve a szabványos C++ kódok fordítására is. De ezekben a Windows rendszer specifikus lehetőségeinek egy részét (többek között a COM objektumokét) nem használhatjuk ki, bár a Microsoft már megkezdte a COM specifikációnak megfelelő rendszerek átírását más (például Unix) platformokra.

A Windows specifikus lehetőségeinek más környezetbe való beillesztése váltott ki vitát a Java és a Visual J++ esetében is. Az utóbbi környezet az alapvetően objektumorientált Java nyelvet használó programozóknak is biztosítja az említett COM komponensek és ActiveX vezérlők használatát. E lehetőség biztosítását indokolja, hogy ameddig a program fejlesztése és futtatása 32 bites Windows rendszerekre korlátozódik, addig a Java előnyeit ötvözni lehet az API hívásokat is használó komponensek sebességnövelő hatásával. Ez azonban a Microsoft által a felhasználók rendelkezésére bocsátott, módosított Java futtatórendszert igényli. A Java ugyanis nemcsak programozási nyelv, hanem más interpretált nyelvekhez hasonlóan futtatórendszer is. Esetünkben ez az interpreter a Java Virtuális Gép (Java Virtual Machine — JVM), a baj csak az, hogy a Microsoft által bevezetett, kibővített Java-megoldások nem tartoznak bele a szabványos és a Sun által lefektetett Java specifikációba. A vita tovább tart arról, hogy a Microsoft-féle rendszer telepítését követően a szabványos vagy a kibővített Java specifikáció legyen-e az alapértelmezett rendszer.

A Java körüli vitától eltekintve a COM specifikáció megalkotásával a Microsoft sokat lendített a Windows platformra írt fejlesztésen. Az ezen alapuló komponensek általában ActiveX vezérlőként jelennek meg a fejlesztés során. Korábban, a 16 bites világban is volt már lehetőség különálló komponensekben leválasztani a szoftver funkcióinak egy részét. Ezeknek a VBX-eknek azonban sok korlátjuk volt. A COM komponensek, köztük az ActiveX vezérlők, fizikai állományként tipikusan .DLL, illetve hasonló felépítésű OCX fájlokban jelennek meg, amelyeket a korábbiakkal ellentétben a főprogram nem a hagyományos .DLL hívási felületen, hanem az erre épülő, objektumorientált alapelveknek megfelelő COM felületen keresztül ér el. Nyilvántartásukat a Windows központi adatbázisa, a Registry végzi, és ezáltal válnak hívhatóvá a gépen telepített bármely program számára.

A regisztrált COM szabványú komponensek a szülő programnyelvtől függetlenül szabadon rendelkezésére állnak a különböző nyelveket támogató, de ezeket a komponenseket használni tudó fejlesztőeszközzel dolgozó programozóknak is. Így a COM komponensekre bontott szoftver egyes komponenseit nem kell az adott nyelven fejleszteni, sőt még az sem követelmény, hogy a Visual Studio valamely programozási nyelvével készüljenek. Maga a Visual Studio egyébként szintén COM komponensekből épül fel, biztosítva ezzel a különböző programfejlesztési eszközök működésének és vizuális megjelenésének egységességét.

A Microsoft-eszközökkel végzett vizuális programfejlesztésben külön jelentőségük van a COM objektumoknak a verzióütközések elkerülésében. Ezek az objektumok, éppen az objektumorientált programozásnak köszönhetően, bármely szülőobjektumuk hívási felületét prezentálni tudják a hívó alkalmazás felé, ellentétben a hagyományos eljáráskönyvtárakkal, amelyek a hívó program irányába csak egyetlen hívási felületet adnak. Ezzel a többarcúsággal a fizikailag lecserélt könyvtár továbbra is kapcsolatot tarthat a korábbi verzió(ka)t igénylő programokkal.

Az, hogy a Visual Studio megtartja a forráskód kompatibilitását a korábbi verziókkal, lényeges lehet az előző eredményekhez való hozzáférés és azok tapasztalatainak felhasználása szempontjából. Ezt olyan elemek is támogatják, mint az automatikus kódkiegészítés és a szintaxisfigyelés. A Visual Basic pedig egyszerű kezelhetősége okán valószínűleg továbbra is az egyik legnépszerűbb tagja lesz a csomagnak.

Simay Endre István

A név kötelez

A Pascal átváltozásai

A 60-as évek második felében egy svájci számítástechnikus, Niklaus Wirth megalkotta első programozási nyelvét, és azt Blaise Pascal matematikusról nevezte el. A nyelv kialakításának egyik fő célja az volt, hogy használóit rászorítsa a strukturált és logikailag áttekinthető programfejlesztésre. Ugyanezt a célt szolgálták Wirth később megalkotott nyelvei is, mint a Modula II és III, melyeknek mind a struktúrája, mind a szintaktikája a Pascalra hasonlít. Áttekinthetősége és az Algol eredetű C-nél könnyebb tanulhatósága következtében a Pascal gyorsan elterjedt a programozók körében, sok géptípusra és operációs rendszerre átvitték. Az így átírt Pascal verziók közül néhány szabadon hozzáférhető, de az IBM szabványú (W)Intel világ személyi számítógépein dolgozó programozók számára a Pascal mint programozási nyelv az évek során összeforrott a Borland névvel.

Már a Borland cég alapítása is a Pascal nyelvhez kötődik. Pontosabban a Turbo Pascalhoz, amely először 1983-ban került ki a fejlesztők kezéből. Philippe Kahn, a Borland International alapítója az általa megvásárolt Kompass Pascal fordítóra alapozva új Pascal programozói környezet fejlesztését kezdte meg. Ebből a felhasználóknak közvetlenül eladható, gépi kódot készítő fejlesztői környezet született.

A maga korában nagy áttörés volt ez a termék. Egyszerű fejlesztői környezetében a Turbo Pascal 1.0 a Pascal fordítón kívül egy WordStarral kompatibilis szövegszerkesztőt tartalmazott, és nevéhez méltóan igen gyorsan fordított korának konkurens termékeihez képest.

Az 1.0-s Turbo Pascalt egy évvel követte annak következő verziója, amely maximálisan 64 KB memóriát használó .COM programok készítésére volt alkalmas, CP/M vagy DOS operációs rendszert használó személyi számítógépeken. Ez azonban az adatterületre nézve nem jelentett szoros korlátozást, mert már a korábbi verzió is támogatta a heap (halom) területen végzett dinamikus memóriafoglalást. Ebben a verzióban jelent meg a forrásszöveg fordításidejű szintaktikai ellenőrzése, amelynek segítségével a fordítóprogram által hibásnak talált kódsort a szövegszerkesztőben azonnal javítani lehetett.

A Turbo Pascal környezetek első nagy ugrása a következő, 3.0-s verziónál következett be, és a PC-s világban talán ez volt az első igazán elterjedt Pascal nyelvű fejlesztőkörnyezet. Hazánkban szintén ez terjedt el, rögtön a megjelenését (1985) követő időben, szinte egyidejűleg az IBM PC-k és klónjaik gyors elterjedésével. E programok memóriahasználatát még korlátozta a 64 KB-os limit, de itt is megvolt a heap dinamikus lefoglalásának lehetősége, ami ezt követően a Turbo Pascal verziók többségének beépített funkciója maradt.

A kód növelése érdekében két megoldást alkalmaztak. Az egyik a láncolás (chain) volt, mellyel bevezették azt a lehetőséget, hogy az adatterület megőrzése mellett újabb programot lehetett betölteni a kódterületre. Ezt a kor néhány másfajta számítógépre írt programkörnyezete, így az Enterprise 128-hoz való IS-Basic is támogatta. A Turbo Pascalban azonban megjelent az overlay (átlapolási) technika támogatása, mellyel csak a kód egy részét kellett a háttértárról újratölteni. A módszer eredményességét jelzi, hogy ez a technika is végigkíséri a további verziókat.

Szintén a 3.0-s verzióhoz jelentek meg a különböző kiegészítő modulok. Ezek egyike a Logo nyelvben használt, és néhány más nyelvbe is beépített „teknőcgrafika" támogatása, mely lehetővé tette a CGA kártyával felszerelt gépeken a grafikus programok fejlesztését. Más jellegű kiegészítések szintén megjelentek a Borland cég által forgalmazott „toolbox" gyűjteményekben, melyek között az adatbáziskezelésben és a játékkészítésben érdekelt programozók egyaránt találhattak munkájukat elősegítő csomagokat. Ezt követően a Turbo Pascal eszközök verzióváltása hosszabb ideig szünetelt, de közben a Borland fejlesztői teljesen átdolgozták a Turbo Pascal fejlesztőeszközt.

Az 1987-ben megjelent 4.0-s verzióban debütált az az új kódszerkesztési és fordítási koncepció, amely később sikerre vitte a Turbo Pascal környezeteket a nagyméretű, komplex programok fejlesztésében is. Ez az újítás az úgynevezett unitok bevezetése volt. Ezek használatával a forráskód (részben a logikai strukturáltságot is követve) fizikailag szétoszthatóvá, modulárissá vált, az egyes unitok mint programmodulok külön-külön kerültek fordításra.

Az unitokban definiált programelemek folyamatosan felhasználhatók a főprogram kódjában, de az ezeket tartalmazó, definiáló forráskód szegmenseket elegendő egyszer lefordítani. Később ez a bináris állomány kerül beszerkesztésre a bináris programfájlba, amely a Turbo Pascal 4.0-nál már .EXE formátumú. Az így szegmentált programkódban az egyes modulok kódjára külön-külön vonatkozik a 64 KB-os korlát, bár a program teljes adatszegmense továbbra is ugyanerre a méretre van korlátozva. További újdonság: a heapkezelő a halomterületen már csak a kívánt blokkméretet foglalta le, így takarékosabb, de lassabb is lett a memóriahasználat (emiatt ezt később nem tartották meg). Szintén itt, a 4.0-ban jelent meg a külső rutinok tárgykódú .OBJ kódból való átvételének a lehetősége, de a 4.0 fordítója már .OBJ állományok készítése nélkül dolgozott, ugyanakkor takarékosan, mert a fordítási folyamatban megjelenő smart linking segítségével csak a ténylegesen használt rutinok kódját fordította be a végleges programfájlba.

Számos újdonsága mellett ennek a verziónak maradtak hiányosságai, nem oldotta meg például a nyomkövetést, a metódus típusú paraméterek átadását, a memóriakezelésben az EMS használatát, ráadásul a fejlesztők megszüntették az overlay technika alkalmazását. Ez utóbbit valószínűleg a unit rendszerre hivatkozva. A későbbi verziók ismét támogatták a rugalmas kódszegmenskezelési lehetőséget. A 4.0-s verziót alig

egy évvel követő 5.0 elsősorban a hiányosságok kiküszöbölését jelentette.

Annál nagyobb újdonság volt az 5.5-ös verzió, melyben a Borland először vezette be az objektumorientált programfejlesztés teljes körű támogatását. Megjelentek az erre jellemző új foglalt szavak: „object", „constructor", „destructor", „virtual". Ehhez szükség volt az 5.0-ban megjelent nyomkövető (Turbo Debugger 1.0) átírására is.

A külső megjelenésében a DOS-os Borland termékeket végigkísérő fejlesztőkeret (IDE - Integrált Development Environment) helprendszere is kibővült. A keretrendszerben teljessé tett objektumorientált felépítésnek köszönhetően megnyílt a lehetőség, hogy a helpállományból kódrészleteket másoljunk át az éppen fejlesztett programkódba, és bevezették a Tour segédprogram használatát, melynek célja szintén az összetett fejlesztői környezet megismertetése volt.

Az ekkorra kialakult fejlesztőkörnyezet stabil és kiérlelt voltát jelzi, hogy később nem csak a Turbo Pascal fejlesztőkörnyezetnek lett sajátja. A Turbo Vision néven közismertté vált keret megjelent az 1990-ben piacra került Turbo C 1.0, majd 2.0 fejlesztőeszközben is, a Tour programmal együtt. A Borland cég a fejlesztőknek rendelkezésére bocsátotta az Turbo Vision keretrendszert, és ezt a „jó szokását" a későbbi fejlesztőeszközökben is megtartotta. A programozók szinte hiánytalanul megkapták mindazokat a programfejlesztési lehetőségeket, amelyek a Borland által is használt IDE-t jellemezték.

A Turbo Pascal 6.0-hoz tartozott a Turbo Vision 1.0-s verziója, de a nyelvi fejlesztés is előtérben maradt, lehetőséget kínálva a programozónak a Pascal forráskódba Assembly nyelvű betéteket is beépíteni (Build-in Assembler – BASM). Tovább lazították a Pascal nyelvi konvencióin, így függvények is meghívhatóvá váltak eljárásként. A lefordított program alkalmazkodni tudott a futtatásra szolgáló gép képességeihez, mert a feltételes fordítás segítségével lehetővé vált a 80286-os és újabb processzorok utasításkészletéhez alkalmazkodó hatékonyabb kód generálása. Megjelent a védett módú parancssoros fordítás lehetősége (TPCX.EXE), mely a későbbi verzióban már a külön fejlesztőkörnyezettel is támogatott védett módú fejlesztés alapját teremtette meg.

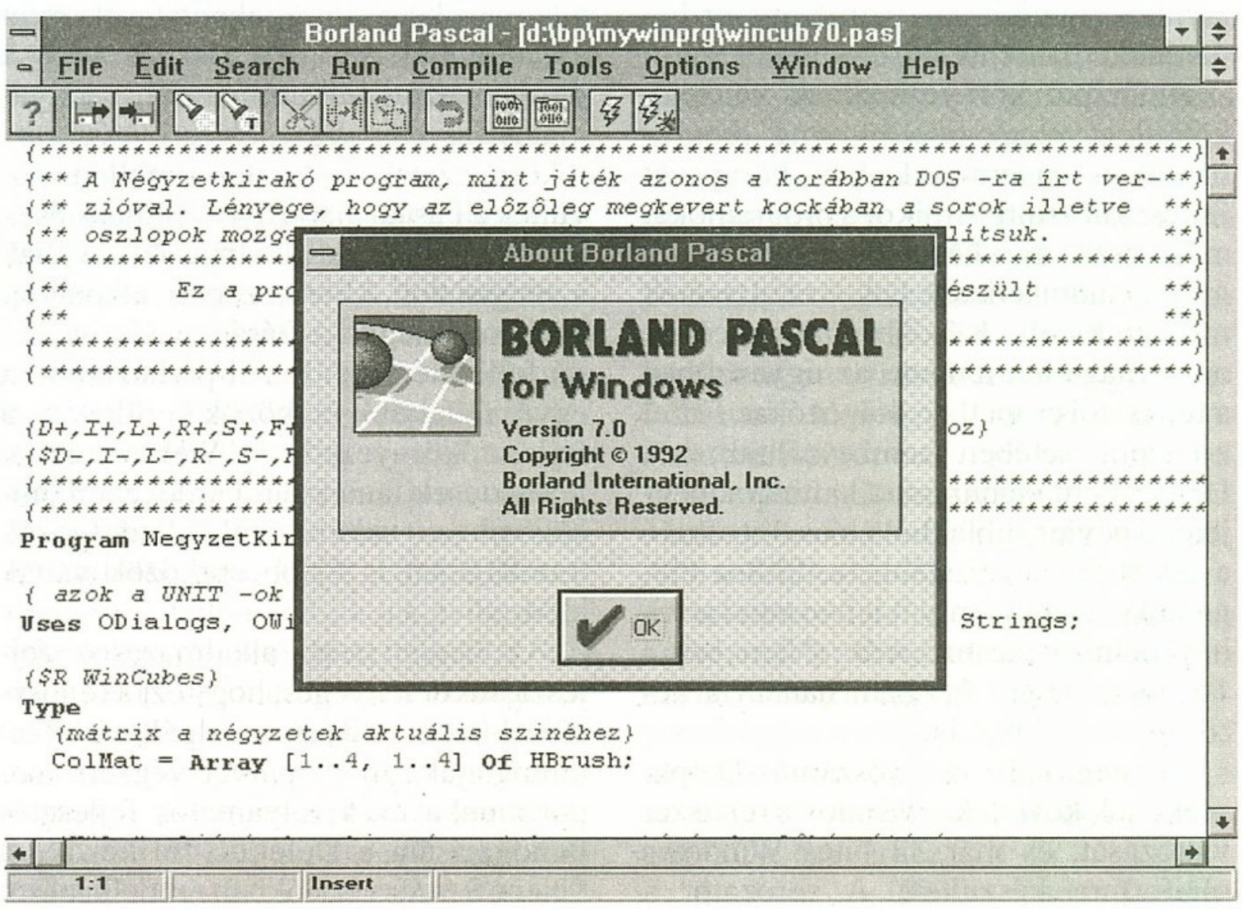
Az említett programok mind karakteres DOS felületre készültek, de nem feledkeztek meg a Turbo Pascallal közel egyidős MS Windowsról sem. A 80-as évek elején jelentek meg a Windows első verziói, az 1.0, 1.02, illetve 2.0. Ezeknek még számos fogyatékossága volt, ami terjedésüket gátolta, és áttörést csak a 3.0, illetve az annak hibáit részben korrigáló 3.1 hozta meg. Hasonlóan a C++ programozókhoz, ekkor a Pascal-hívők körében is felmerült az igény a Windows felületen futtatható programok írására. Azonban a DOS-os Pascal fejlesztőeszközök, ellentétben például a Borland C++ 2.0-val, még korlátozott mértékben sem tették lehetővé windowsos programok fejlesztését. Emiatt a Turbo Pascal 6.0 megjelenését követően hamarosan elkészült a Turbo Pascal for Windows 1.0, majd az 1.5. A unitok fejlécében látható a „TPU9" bejegyzés, mely jellemző a DOS-os Turbo Pascal 6.0 fordítójára. Ezekben a programkörnyezetekben a Windows környezeteknek megfelelően át kellett alakítani a nyomkövetést is, de a hibakereséshez nem készült a fejlesztőkörnyezetbe integrált debugger, hanem külső programot kellett hozzá használni (Turbo Debugger for Windows).

A Turbo Pascal 6.0 korszak kettőssége a következő verzió kiadásakor megszűnt. A DOS-os programozók számára megjelent a Turbo Pascal 7.0, de szinte ezzel egyidejűleg (1992 októbere) a Borland Pascal with Objects 7.0 programcsomag is. Ez utóbbi tartalmazta a valós (real), illetve a védett módú DOS programok fejlesztésére alkalmas Turbo Pascal 7.0 programcsomagot, de a teljes értékű 16 bites windowsos fejlesztőkörnyezetet is.

A DOS-os Turbo Pascal 7.0 tartalmazza a Turbo Vision új verzióját és a szükséges javításokat. A Borland Pascal with Objects a védett módú programozáshoz azonban több újdonságot is bevezetett a DPMI (Dos Protected Mode Interface) alatt futó alkalmazások fejlesztéséhez. Itt a programozó dinamikusan kapcsolható könyvtárakat (DLL) alakíthat ki, melyek a WINAPI unit felhasználásával a DOS védett módjában és Windows platformon egyaránt használhatók.

A Windows platformra végzett fejlesztésben a legnagyobb áttörést a Pascal nyelvet is tovább bővítő Delphi jelentette. Ennek a fejlesztőkörnyezetnek már 16 bites első verzióját is kifejezetten erre a platformra fejlesztették. Bár néhány egyszerű fogással a Delphi 1.0 Pascal-fordítójával (DCC.EXE) is lehetőség van DOS programok parancssoros fordítására. A fejlesztőkörnyezetből azonban nem, mert a szükséges beavatkozásokat lényeges rendszerkönyvtárakban kell megejteni.

A Delphi megjelenésével alaposan átalakult a Pascal alapon végzett windowsos programozás. A programozó válláról a gyári objektumkönyvtárak leveszik a legtöbb aprólékos feladatot, melyekre korábban nagyon oda kellett figyelni, mert könnyen ki lehetett siklatni a Windows üzenetkezelő rendszerét, működésképtelenséget, rendszerlefagyást okozva. A kiterjesztett objektumrendszer alapján a Delphiben megnyilvánuló Pascal implementáció az objektumorientált programozást szintén támogató korábbi verziók bármelyikénél jobban rászolgált az Object Pascal elnevezésre, bár fejlesztőkörnyezete alapján akár Visual Pascalnak is nevezhetnénk.



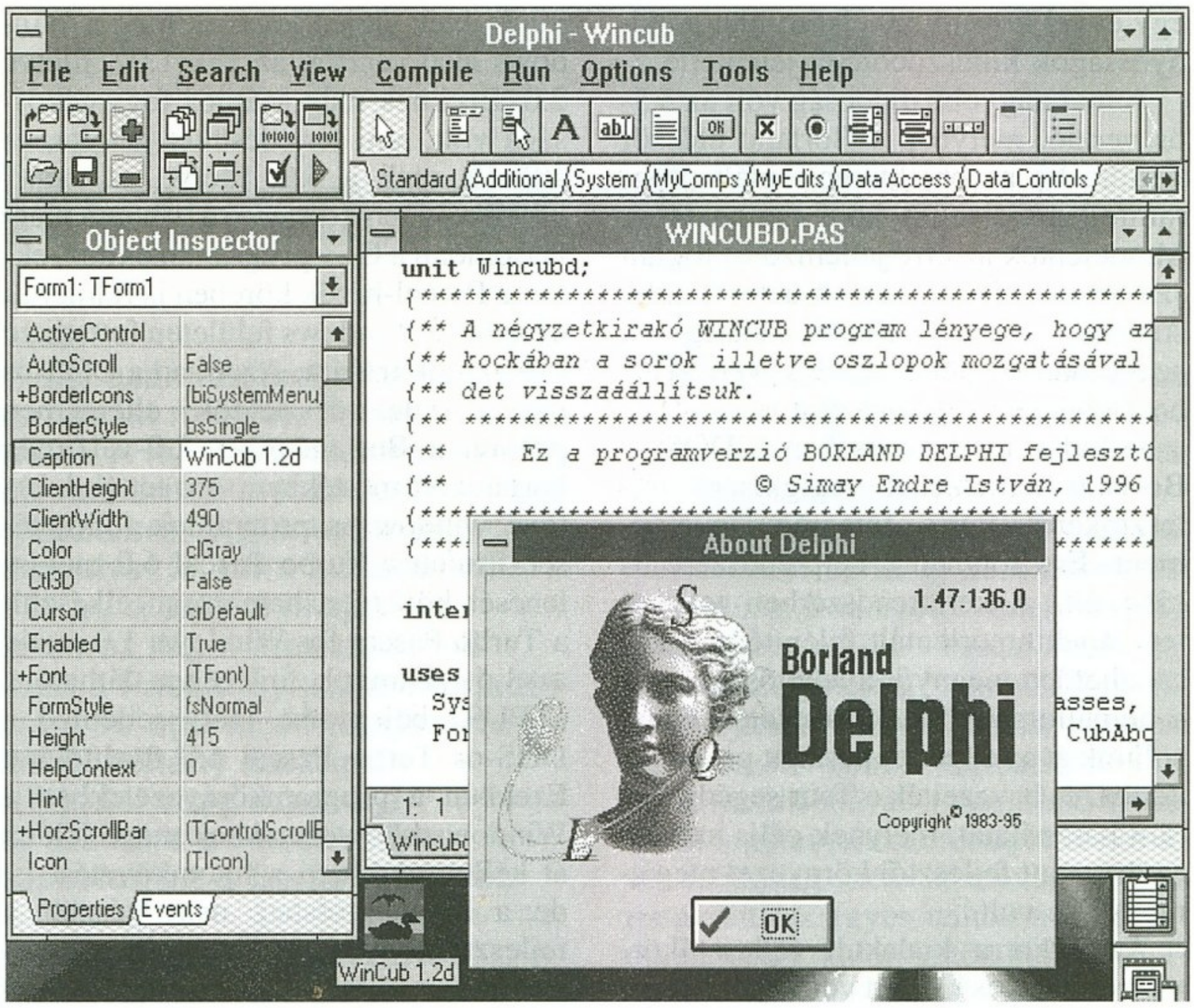
A Delphiben a gyakorlati programozás során a fejlesztő előtt szinte kizárólag a vizuális komponensek jelennek meg, melyekkel meghatározható, hogy mi történjen egy gomb lenyomásakor vagy egy adatbáziscella tartalmának megváltoztatásakor. A fejlesztőmunka során természetesen továbbra is szükség van programozási ismeretekre, annál inkább, minél finomabbra kívánjuk hangolni az adott alkalmazás működését. Ha pedig a programozó nem elégszik meg a gyári vagy a más forrásból származó komponensek által nyújtott választékkal, új komponenseket készíthet, ami persze nagyobb felkészültséget követel tőle.

A komponensek készítésekor felmerülő problémákkal, az objektumhierarchia beható megismerésének igényével, a Windows API függvényeinek kezelésével a fejlesztőeszköz használói nem minden esetben szembesülnek. Elegendő, ha a vizuális komponensek palettájáról kiválasztott komponenst elhelyezik az alkalmazás ablakát reprezentáló formra, majd meghatározzák a főbb jellemzőket és a kezelési reakciókat. Ehhez már a Delphi 1.0 fejlesztőrendszer is kifinomult lehetőségeket adott. Az említett komponenspalettán kívül állandóan kézre esik a komponensek jellemzőit elérhetővé tevő Object Inspector, és lehetőség van a DOS-os verziókban megszokott programkövetési megoldásokra is.

Magát a fejlesztőeszközt is kedvünkre alakíthatjuk. Akár azt is beállíthatjuk, hogy a Borland cég korábbi termékeiben (Turbo Pascal, Turbo C) megszokott billentyűkezelést kívánjuk-e használni. A szerkesztés színkombinációit (syntax highlighting) szintén szabadon paraméterezhetjük.

A Delphi környezeteknek van egy különlegessége, amely nem a programozókat, hanem a Delphi környezet fejlesztőit érinti. Amikor a programokat még csak egy-két ember készítette, mindig tudni lehetett, hogy mely termék mögött ki áll. Később a nagy cégek neve maga alá temette az ugyanabban a fejlesztői csapatban dolgozókat. Ezzel a Delphi esetében szembeszálltak, és a Help/About menüpontra kattintva megjelenő névjegyablakból előcsalogatható a teljes fejlesztői stáblista. Ehhez elegendő az Alt billentyűt lenyomva tartva begépelni a „team" szót. (Nemcsak a 16 bites Delphi 1.0-ban, hanem a későbbi verziókban is.)

A magasabb verziószámú Delphi eszközök követték a Windows rendszer változását, és már 32 bites Windows platformra készültek. A sorozatot a



Delphi 2.0 nyitotta meg. Ez a környezet mind megjelenésében, mind képességeiben a Delphi 1.0 32 bites átirata. Mégis rendhagyó verzió, mert a Borland ekkor kezdte meg a Pascal és a C++ alapú fejlesztőeszközök közös arculatra formálását Windows alatt is. Ennek első példája a Borland C++ Builder 1.0 volt (1997), de a C++ Builder további verziói rendre követték a megfelelő Delphi verziókat (3.0, 4.0), azonos lehetőségeket biztosítva a különböző nyelvekkel dolgozó programozóknak.

Az említett magasabb verziójú Delphik fejlesztőkörnyezetében megjelentek olyan kényelmi szolgáltatások, mint az automatikus kódkiegészítés, vagy a 4.0-ban az objektumhierarchia folyamatos figyelése és kijelzése. A fejlesztőkörnyezetek fejlesztése mellett bővültek a Pascal alapú nyelvi fejlesztések is. Az új verziók alkalmasabbá váltak a megosztott kliens/szerver alapú fejlesztések megvalósítására, a Corba alapú fejlesztésekre. Ezzel párhuzamosan egyre fejlettebb eszközök kerültek be a fejlesztőkörnyezetbe a Webre történő fejlesztések támogatása érdekében, miközben az adatbázisokkal dolgozók munkájában is újabb eszközök váltak elérhetővé.

Az összetettebb alkalmazások fejlesztésekor lényeges, hogy ezt a fejlesztőeszközök milyen szolgáltatásokkal támogatják. A Delphivel végzett csoportmunka és a folyamatos fejlesztés támogatására a Delphi 3-tól kezdve a Client/Server verziókban rendelkezésre áll a PVCS Version Manager. Szintén ettől a verziótól teljes a 32 bites Windowsban alkalmazott COM alapú fejlesztések támogatása. Ezt felhasználva meglévő komponenseinkből készíthetünk például más fejlesztőeszközökkel dolgozók számára hasznosítható ActiveX vezérlőket, de fel is használhatjuk a mások által készítetteket. Ez utóbbi támogatásához tartozik, hogy a Windowsban regisztrált objektumok számára a Delphi automatikusan képes illesztőegységet (megfelelő unit .PAS forrást) generálni.

Ezzel és az OLE objektumok általában is egyszerű kezelésének lehetőségével a fejlesztő a Windows rendszerrel szorosan integrált alkalmazást készíthet. Akár közvetlenül is adatokat cserélhet más OLE alkalmazásokkal, például a Worddel, Excellel. De az ActiveX használata teszi lehetővé olyan eszközök elérését az újabb Delphi és C++-Builder verziókkal készült alkalmazások számára, mint az adatbázis-használatot megkönnyítő ADO (AxtiveX Data Objects), vagy a multimédiás lehetőségeket biztosító DirectX rendszer.

A Borland a fejlesztőeszközök terén azonban nemcsak a Wintel rendszereket célozta meg. Olyan programozói eszközök átiratát készítették el AS/400-ra, mint a Delphi, a C++ Builder és a JBuilder. Ez utóbbi tisztán javás fejlesztést tesz lehetővé a Delphire emlékeztető környezetben. Akár más verziójú (JDK 1.1x, 1.2) futtatómotorokkal is.

Simay Endre István

Visual FoxPro 6.0

Róka fogta xBase

Az adatbáziskezelők egyik legrégebbi csoportját képviselik a dBase-hez hasonló programozási nyelvet és adatbázisformátumot (*.DBF) használó, XBase gyűjtőnévvel jelölt programok. E csoport további közös vonása, hogy időről időre megkongatják felette a vészharangot. Már többen és többször ki- és bejelentették, hogy e nyelveken és e nyelvekhez fejleszteni energiapocsékolás, mert hamarosan kihalnak, és átadják helyüket a többi relációs adatbáziskezelő rendszernek. Azután mégis kijönnek az újabb XBase alapú fejlesztőeszközök, újabb nyelvi elemekkel. (A Visual dBase 7.0 ismertetése az Új Alaplap 1998. márciusi számában olvasható.)

Az új xBase fejlesztőkörnyezetek közé tartozik a Microsoft által fejlesztett FoxPro legújabb verziója, a Visual FoxPro 6.0 is, mely része a több más fejlesztőeszközt is tartalmazó MS Visual Studio csomagnak. Futtatókörnyezete a 32 bites Windows platform, hardverigénye sem tér el lényegesen a többi hasonló célból írt programtól. Windows 95 OSR1 esetében már 16 MB RAM-ot tartalmazó gépen is probléma nélkül fut, bár ilyenkor bele kell törődni lassúságába. Minimális telepítéskor 20 MB merevlemez-területtel is megelégszik, de ilyenkor a program nagy része a CD-n marad, és a megfelelő működéshez több memória és gyors CD-olvasó kell. Tipikusnak tekintett telepítési opciót választva a merevlemezigény 100 MB, teljes telepítéskor 240 MB körüli. De ebbe beletartozik a száz megabájt nagyságrendű Online súgóadatbázis is.

A munka kezdetén a FoxPro embléma után bejelentkező első kép egyben egy indító varázsló, melynek későbbi megjelenését le is tilthatjuk. Itt kérhetjük az Online súgót, a meglévő projektek megnyitását vagy újak kezdését is. Az utóbbi esetben a Visual FoxPro a főablakban egy üres vázat nyit meg, amely mellett megtaláljuk a parancsablakot is (Command), arra az esetre, ha parancsok kiadásával dolgozunk.

Az üres váz folyamatosan tölthető fel a különböző egységekkel. Ezek létrehozását varázslók segítik. Új adattáblák létrehozásakor előre definiált sablonok közül is választhatunk, amelyekből a mezők módosításával könnyen kialakíthatjuk a saját célunknak legjobban megfelelő táblázatvázat. Ahogy egy xBase rendszerben azt megszoktuk, szükség van a táblázat indexelésében részt vevő kulcsmezők megadására, majd a varázslóval végzett táblázatkészítés lezárásaként elmenthetjük az üres táblát, de tovább is dolgozhatunk vele. A későbbi módosításokat parancsablakból vagy menüből kezdeményezhetjük.

Az adattábláinkkal végzett munkához több lehetőség közül választhatunk. Konzervatívabb felfogásban (vagy ha arra van szükség) az adattáblák közötti kapcsolatok beállítására többé-kevésbé alkalmazhatjuk a 16 bites FoxPro for Windows 2.x verziókban is már ismert megoldásokkal analóg utat. Lehetőség van azonban külön nézetek létrehozására is. Ezek egyfajta átmenetet jelentenek az említett 2.x verziók nézetablaka (View) és a lekérdezések között. Létrehozásukat szintén varázsló segíti, melynek végén az elmentéssel párhuzamosan az új tervezőbe (View Designer) is kérhetjük magunkat.

A tervezőben újabb adattáblákat állíthatunk be, definiálhatjuk az adattáblák közötti kapcsolatokat, lekérdezéseket, mely utóbbiak lefuttatását a jobb egérgombbal előcsalogatható lebegőmenüből is kezdeményezhetjük. A lekérdezések SQL kódját is innen tekinthetjük meg, és különböző kimeneteket definiálhatunk (SQL kurzor, adattábla generálása, a jelentések, címkék készítése stb.). Felhasználhatjuk a korábbi verziókban és más programokkal készült DBF formátumú adatbázisokat is.

A kompatibilitás szintjét a programmal kapott dokumentáció lefelé a 3-as verzióig említi, de az általam korábban a 2.5-ös verzióban készült egyszerű próbaalkalmazás elemeivel sem volt sok probléma. A legtöbb más fejlesztőeszközhöz hasonlóan a FoxPróra is igaz, hogy a verziók közötti átjárhatóság akkor nehéz, ha nagy a verziók közötti időrés, vagy ha az adott alkalmazás egyes verziók képességeire van kihegyezve.

A konverzió az említett távoli verziók között az adattáblák és lekérdezések esetében általában nem okozott gondot. A jelentéseknél már előfordult, hogy a kiértékelés átvételében kisebb pontatlanságok voltak, azonban a módosítások könnyen végrehajthatók. Érdemes tehát a konverziókat megpróbálni. Sikeres konverziók esetén is célszerű viszont új képernyőket (formokat) alkotni.

Ezek létrehozását szintén varázsló segíti, amelynek működési eredményeként kapunk egy alapablakot a kiválasztott mezőkkel és egy navigációs gombsorral. Ezt azután tovább szerkeszthetjük, megváltoztatva a meglevő elemek tulajdonságát, vagy újabb elemeket helyezve az ablakba. Ezek közé egyaránt felvehetünk korábban elkészített saját FoxPro és a rendszerbe installált ActiveX vezérlőket.

Az elemek viselkedésének szabályozásához továbbra is lehetőség és szükség van bizonyos programozásra. Ez lehet a megjelenő grafikus elemek tulajdonságainak megváltoztatása is, hogy alkalmazásunk interaktivitását fokozzuk. De lehet a teljes alkalmazás szintjén ható programokat is készíteni, ahogy azt korábban is megtehettük. Az új Visual FoxPro vizualitását kihasználva gyorsan elkészíthetjük alkalmazásunknak a felhasználóval kapcsolatot tartó elemeit, melyeket azután projektbe épített programokkal egységbe foglalhatunk. Akár egyedi gépen, akár hálózatban dolgozva.

Komplett alkalmazások összeállításánál kihasználhatjuk a már említett ActiveX integrációk azon előnyét, hogy az adateléréshez is lehetőségünk van külső, más típusú adatbázisok elérésére az ADO segítségével. Hálózaton dolgozva így nem szükséges a különböző adatbázisokat „közös DBF nevezőre hozni”, és csatlakozhatunk egy Microsoft SQL Serverhez is.

Simay Endre István

Integrált Java fejlesztőeszköz

IBM VisualAge for Java v2.0

Nemrégiben megjelent az IBM VisualAge for Java fejlesztőeszköz 2.0 verziója, amely híven követi a VisualAge família hagyományait, természetesen tökéletesítve az eszközt, és ezzel könnyítve a fejlesztők munkáját. Három verzióban is hozzájuthatunk, és ezek közül a legkisebbet, az Entry Editiont ingyen is birtokba vehetjük (persze csak akkor, ha kipróbálási szándékunkhoz a maximum 500 osztály kezelését megengedő korláttal szembeni tolerancia is társul). A két kereskedelmi verzió a Professional és az Enterprise neveket kapta, utalva a megcélzott felhasználói csoportokra.

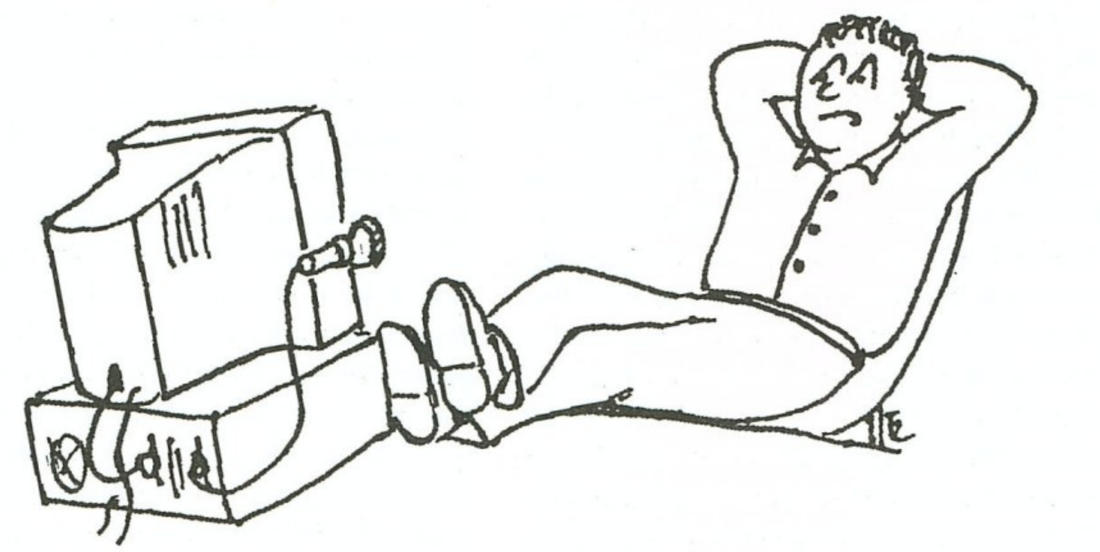
Aki még nem próbálta ki a VisualAge fejlesztőeszközt, viszont jártasságot szerzett más neves gyártók hasonló célra szánt termékeinek használatában, annak ajánlani tudom a szoftver kipróbálását, mert sok tekintetben érdekes és tanulságos kirándulásban lesz része. Az alábbiakban sorra veszem a sajátosságok és szolgáltatások sokaságából azokat, amelyekkel tapasztalataim szerint más termékek nem (vagy csak kezdetleges szinten) rendelkeznek.

— Az első és legfontosabb jellemző, hogy nincsenek fájlok, így nincsenek többé elveszett fájlok sem, mert a környezet egy kicsiny adattárat (repositoryt) használ forrásaink tárolására (bár ebben a verzióban már lehetőségünk van törölni is belőle). Ebbe az adattárba kerül be mindaz, amit az eszköz használója a telepítés utáni pillanattól kezdve megalkot, legyen az akár az első Hello World alkalmazás, vagy aktuális munkája. Az idő múltával ez tekintélyes méretűvé dagadhat, ezért hogy ne kelljen a felgyülemlett munkák tömegében folytonos keresgéléssel töltenünk napjainkat, az egész fölé egy Workbench elnevezésű grafikus felület került, amelyben mi magunk határozhatjuk meg, hogy az adattárban tárolt anyagok mely elemeit kívánjuk látni és használni munkánk során, és melyek azok, amelyekre a pillanatnyi fejlesztés közben nem lesz szükség. Természetesen ez utóbbiakat is könnyen felhelyezhetjük a Workbench felületére, vagy a menet közben fölöslegesnek találtakat eltávolíthatjuk onnan. Lehetőségünk van megtekinteni az ily módon kiválasztott projekteket, package-eket és osztályokat hierarchikus elrendezésben, package-enként vagy osztályonként, majd ezek közül kiválasztva az érdeklődésünknek megfelelőt, megnyithatjuk azt egy újabb ablakban akár szövegszerkesztésre, akár — ha az illető osztály vizuális komponens — vizuális szerkesztés céljából. Ha nem akarjuk a desktopunkat ablakokkal elárasztani, akkor a Workbenchben is elvégezhetjük a szövegszerkesztést.

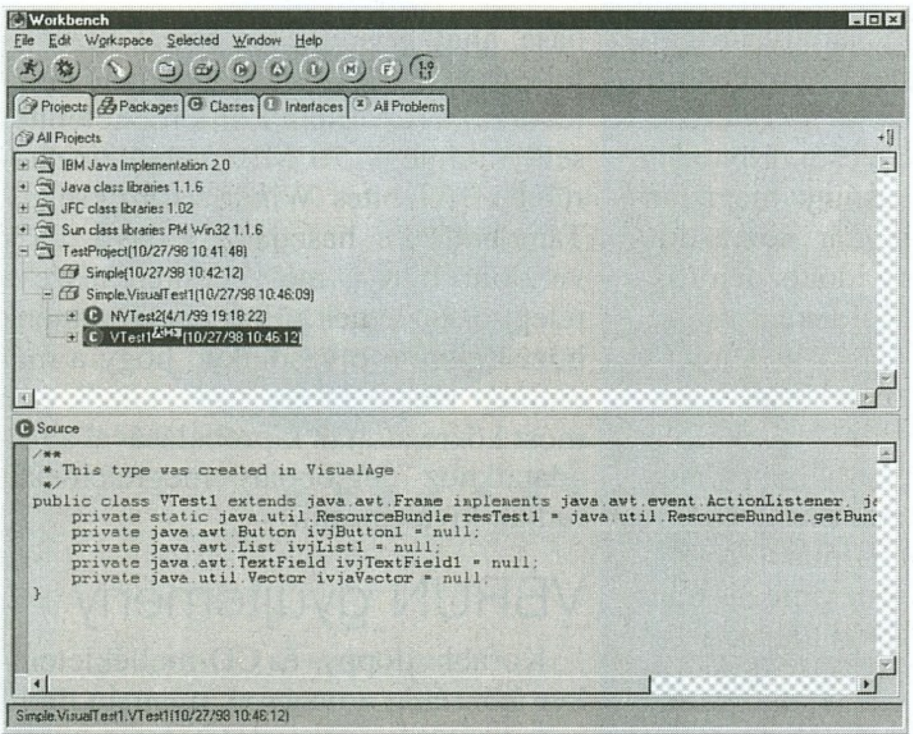
— Az adattárban történő tárolás másik előnye, hogy lehetőségünk nyílik munkáinkat verziószámmal ellátni, megóva azokat a figyelmetlenségből elkövethető módosításoktól, így azok eredeti állapotát mindig elő tudjuk állítani. Ha egy osztályt, package-et vagy projektet lezárunk (új verziószámot rendelünk hozzá), akkor a lezárt forrásainkban történő összes újabb változtatás már az automatikusan megnyíló következő verzióban történik. A lezárt verziók tartalmát a későbbiekben nincs módunk megváltoztatni, viszont a lezáratlan verzió tetszőleges elemét visszaállíthatjuk valamelyik lezárt verzióbeli állapotra. Más fejlesztőeszközöknél ez csak egy külső verziókezelő eszköz használatával oldható meg.

— Külön cikket érdemelne a VisualAge for Java vizuális szerkesztőfelülete. A felhasználói felületek komponensekből történő összerakásában társaival azonos kényelmi szolgáltatásokat nyújt (hiszen ezen a területen sok újat mostanában nem találtak ki), de ezen túl biztosítja az abban elhelyezett logika vizuális létrehozását, megjelenítését és szerkesztését, vagyis nem csupán a felhasználói felület vizuális és nem vizuális elemei jelennek meg a fejlesztő előtt, hanem azok kapcsolata és viselkedése is. A szerkesztés befejeztével egyetlen gombnyomásra generálja a hozzá tartozó forráskódot, amely a többi eszköztől eltérően nem jelenti Adapter vagy Listener osztályok megjelenését, minden logika megfér az eredetileg létrehozni kívánt JavaBean osztályban. A generátor végterméke egy igen egyszerű, rövid és érthető kód. Olyannyira az, hogy ha nem tudnánk az eredetéről, azt gondolhatnánk, hogy emberi kéz alkotta, és igényes programozót sejtenénk mögötte.

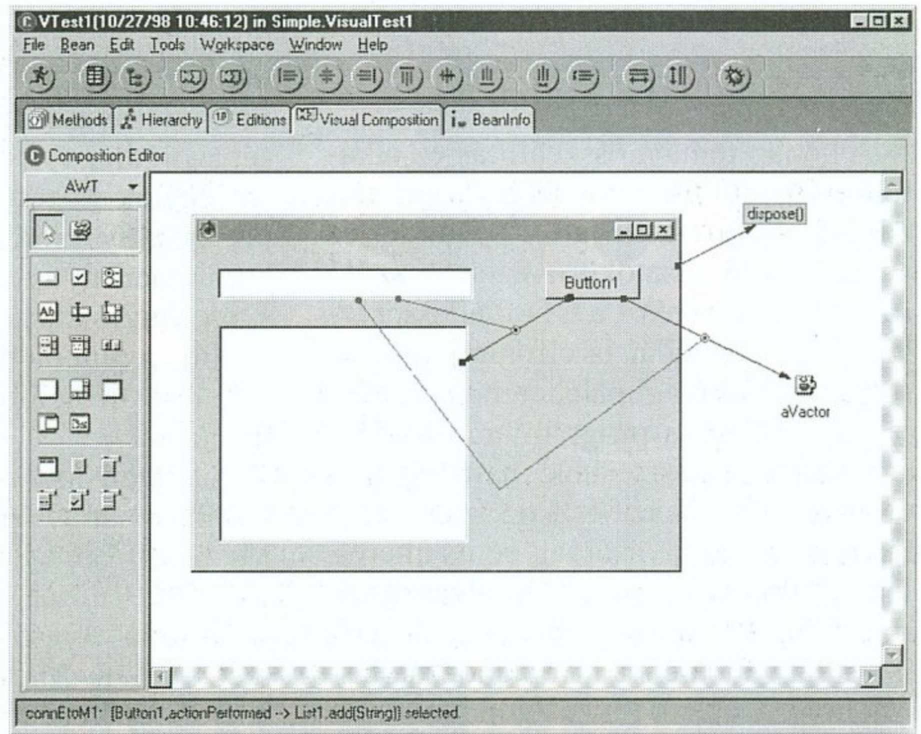
— Következő érdekessége ennek a fejlesztőkörnyezetnek a programban előforduló hibák felderítése. Mivel nin-



— És most az A mátrix elemeire kérek maximum- és minimumkeresést!



1. kép. Workbench



2. kép. Vizuális szerkesztőfelület

csenek fájlok, ezért fejlesztés közben a mentés metódusonként történik, és a mentés folyamata együtt jár a változás azonnali fordításával is. Így amennyiben hibát ejtettünk a forrásban, azt a környezet azonnal észreveszi, és közli velünk, amint elhagyjuk a szerkesztett metódust. Itt nemcsak szintaktikai hibákra kell gondolnunk, hanem a hibát jelző kis X megjelenik minden osztály és annak metódusa mellett, ha történetesen egy olyan metódust vagy attribútumot töröltünk, amelyre más metódusok hivatkoznak. A

Workbench külön szolgáltatást kínál az összes hibás osztály és azok metódusainak egy felületen történő megtekintésére, és a hibák akár helyben, akár az osztály önálló ablakában történő kijavítására.

Első próbálkozásaim során furcsállottam, hogy minden projektet, package-et, osztályt, metódust és attribútumot egy varázsló segítségével kellett létrehoznom, de a későbbiekben ezt megszoktam, így kezdeti idegenkedésem is megszűnt. A varázslók léte és az eszköz által felkínált felület inkább az átgondolt programozást kedvelőknek előnyös, de a kezdők is örömüket fogják lelni benne, feltéve, hogy le tudnak mondani a forráslistákat megjelenítő ablakok kusza tömkelegéről (bár ezt a megközelítést is támogatja) és az azokban való véletlenszerű és esetenként tévedésből vagy kapkodásból elkövetett változtatásokról.

A VisualAge for Java kategóriájában egyedülálló szoftver, amely működésében és szolgáltatásaiban gyökeresen eltérő megoldásokat tartalmaz más hasonló célú termékekkel összehasonlítva, és ezek a különbségek előnyére szolgálnak. Akik először találkoznak vele, azoknak ajánlom, hogy indulásként csinálják végig a helpben található To-Do List példát, melynek során az érdeklődők elegendő ismeretet szerezhetnek a fejlesztőeszköz további lehetőségeinek feltárásához és kipróbálásához a megadott lépéseket végrehajtva.

Mindezen tapasztalataimat az első kísérleteim alkalmával egy 166 MHz-es, 64 MB RAM-mal felszerelt gépen szereztem. Ilyen kiépítettségű gép elégségesnek bizonyult a program kipróbálásához, de azoknak, akik komolyabb projektek fejlesztésébe akarnak kezdeni, inkább PII kategóriájú gépet ajánlok, még ha a fent leírt konfigurációval is megoldható a feladat.

Akikben sikerült felkeltenem a szakmai kíváncsiságot vagy a kísérletező kedvet, azoknak az IBM hivatalos VisualAge for Java webhelyét (http://www.software.ibm.com/ad/vajava/) ajánlom a további ismeretek megszerzéséhez, vagy az ingyenes Entry változat letöltéséhez.

Lovas János

A VisualAge szolgáltatásai

Professional változat

— Data Access Beanek adatbázisok JDBC-n keresztüli egyszerű eléréséhez.
— Integrált fejlesztői környezet, benne
— hibakereső, osztályböngésző,
— vizuális szerkesztő.
— Open Tools Integrator API a VisualAge for Java környezet kibővítéséhez.
— Kapcsolódási lehetőség a VisualAge TeamConnectionhöz, az IBM konfiguráció kezelőjéhez, vagy
— ClearCase-hez, PVCS-hez.

Enterprise változat

(A Professional változat szolgáltatásai mellett.)

— Enterprise Access Builder az IBM CICS (ECI, EPI) tranzakciókezelőjéhez.
— Automated Object to Relational Mapping Enterprise Java Beans alapú perzisztens objektumok készítéséhez.
— Támogatás a TXSeries, SAP R/3 és CORBA fejlesztőknek.
— Visual servlet builder.
— Team programming for Java a fejlesztők munkájának összehangolására.
— Enterprise Toolkits szerveroldali megoldások készítéséhez AS/400-ra és S/390-re.
— Nagy sebességű fordító.
— Távoli hibakeresés.

A hónap témájához

A hónap témája összeállítás ez alkalommal átnyúlik egyrészt a Programozástechnika rovatba (a C++ Builder és a Fortran fejlesztőkörnyezetről szóló írásokkal), másrészt a CD-melléklet tekintélyes hányadát is elfoglalja, olyan sok érdekes anyagból lehetett válogatni.

Peter Cellik programjára például azért esett a választásunk, mert a Sphinx C-- nyelv és fejlesztőeszköz az Új Alaplap rendszeres olvasóinak nem ismeretlen (1996. novemberében olvashattak róla először), másrészt mert más cégek is példát vehetnének arról a hozzáállásról, hogy a szerző tokkal-vonóval (azaz forráskóddal együtt) publikálta programját. A nyelv jelenlegi állapotában 16 bites COM programokhoz és egyéb (például C és Pascal) fejlesztésekhez használható objektumfájlok (.OBJ) készítésére alkalmas. Abban a körben különösen jól alkalmazható, ahol még szükség van a tömör formában elkészítendő programokra. Például alig több mint 2 KB-os PCX-megjelenítő is fordítható a példaprogramokból. Ha pedig valaki bővíteni, fejleszteni szeretné a rendszert, például más forrásokból származó modulokkal és a windowsos fejlesztőkörnyezettel, az megteheti. Ez annak köszönhető, hogy amikor a rendszer eredeti szerzője saját fejlesztőmunkáját egy ideig szüneteltette, akkor a teljes rendszer forráskódját publikálta, beleértve a fordítót, a Turbo Vision alapú környezetet is.

A CD-melléklet GNUC könyvtárában található a DJGPP nevű 32 bites C/C++ fejlesztőrendszer. Futtatásához legalább 386-os processzor és DOS operációs rendszer szükséges, de működik különféle (16 és 32 bites) Windows verziók és OS/2 alatt, sőt DOSEMU-ban is. A DJGPP támogatja a C, C++, Objective C, Bison (yacc klón) és Flex (lex klón) programozási nyelvet, és egy assembler is rendelkezésünkre áll. A programozók választhatnak a jó öreg EMACS szövegszerkesztő és GNU debugger, valamint a RHIDE nevű integrált fejlesztőkörnyezet között. Ez utóbbi a Turbo és a Borland C fejlesztőkörnyezetéhez nagyon hasonló felületet biztosít. A telepítés előtt mindenképpen ajánlatos elolvasni a READ-ME.1ST fájlban lévő útmutatót, a későbbiek során pedig a különféle formátumú bőséges dokumentáció segíthet a programok használatában.

A Delphiben programozók ismét tekintélyes csokrot találhatnak a különböző komponensekből. Köztük számos olyan is helyet kapott, amely programozási illesztőfelületek (Application Programming Interface — API) lefedését szolgálja. Az ebbe a csoportba nem sorolhatók között olvasóink megtalálják néhány korábban már közreadott komponensgyűjtemény időközben frissített verzióját, hogy példaként csak a Prolib gyűjteményt vagy a DirectX kezelését szolgáló komponenseket említsük.

A DirectX programozása iránt érdeklődőknek más ajánlatunk is van. A Microsoft anyagait tartalmazó könyvtárban a multimédiás programozás iránt érdeklődő olvasóink megtalálják a DirectX SDK (Software Development Kit) legújabb, 6.1-es verzióját.

Ugyanitt található meg a Microsoft Java Virtuális Gépre (Java Virtual Machine — JVM) programozóknak szánt fejlesztőcsomag (SDK). A korábbi megoldásoktól eltérően ebben nem egyetlen készlet végzi el a teljes telepítést, hanem a JVM-et és az SDK többi részét külön-külön kell installálni. Ez a megosztott telepíthetőség jól jön akkor, ha csak a JVM frissítését akarjuk elvégezni a korábbi verziókban előfordult hibák kijavítására.

Vannak persze, akik Windowson is más Java eszközöket használnak. Nekik szánjuk a Sun Java eszközeinek új verzióit. A korábbi JDK 1.1.x sorozat folytatásaként kiadott Java Development Kit 1.1.8-at valamint az 1.2-es Java platform frissítését és a csomagok dokumentációit. CD-mellékletünkön az IBM Java eszközeit használók is megtalálják az IBM-JDK legújabb, 1.1.7-es verzióját.

A Java nyelv iránt érdeklődők figyelmébe ajánljuk a különböző Java fejlesztőeszközökről szóló összeállítást. Ez az anyag nemcsak a nyomtatott lap terjedelmi korlátai miatt került a CD-re, hanem azért is, mert így a szövegben előforduló hivatkozásokat (linkeket) közvetlenül elérhetik azok, akiknek gépén megvan a Webre való kilépés lehetősége. Élő Internet kapcsolat mellett olvasva a CD-t, rögtön végiglátogathatók a hivatkozott webhelyek.

A hónap témájához kapcsolódó összeállításban szereplő eszközök közül többet már korábban is bemutattunk CD-mellékleteinken (JBuilder 2, Visual J++ 6.0, IBM VisualAge for Java 2.0 for Windows). Két eszköz kipróbálható verziója azonban a jelenlegi CD-n kapott először helyet, ezek egyike a IBM VisualAge for Java 2.0 OS/2-re írt verziójának próbaváltozata, mely az 1998. novemberi számban közzétett Win32-es verzió párja. A másik a Sybase által gondozott PowerJ vizuális fejlesztőeszköz szabadon elérhető tanulóverziója (Learning Edition). Telepítéséhez mintegy 70 MB merevlemezterület és 32 bites Windows szükséges. Támogatja és használja a Java 1.1.x verziójú JDK-t, mely szintén része a telepítőkészletnek. Ezzel is szeretnénk hozzásegíteni olvasóinkat, hogy a már említett vizuális fejlesztőeszközök és a most közreadottak kipróbálásával a feladataikhoz legjobban megfelelőeket tudják kiválasztani.

VBRUN gyűjtemény

Korábbi floppy- és CD-mellékleteinken közzétett anyagaink nyomán többször kaptunk visszajelzést, hogy a szabadon elérhető, Visual Basic-ben fejlesztett programok csomagjaiból hiányoznak egyes állományok. Jellemzően a VBRUN könyvtár aktuális verziói, mert azokat a programok terjesztői (vélhetően terjedelmi okokból) nem mindig mellékelik, bár maga a fejlesztőeszköz (a 32 bites 5.0 és 6.0 biztosan) lehetőséget ad komplett telepítőkészletek előállítására. Most segítségképpen összegyűjtöttük ezeket az állományokat, kezdve a 16 bites Visual Basic 1.0-tól (vbrun100.dll) a 32 bites VB 6.0-ig (Vbrun60.exe). Ez utóbbinak saját installáló állománya van.

NeoPlanet

Előző számunkban már bemutattuk az Internet Explorerre kezelőfelületet, héjat húzó NeoPlanet programot. Időközben ennek is megjelent a frissített verziója, de a program fejlesztői elkezdték kinyújtani kezüket a Gecko (Netscape Navigator 5.0) felé is. Ennek szabad tesztelésre szánt anyagát (Technical Preview) szintén elhelyeztük korongunkon.

Linux, StarOffice

Előző számunkban azt ígértük, hogy a StarOffice telepítésével és installálásával kapcsolatban sokak által kért eligazítást itt, a CD-Kalauzban tesszük közzé, de az terjedelmileg és tematikailag jobban illett a Fogódzóba (46. oldal). Ugyancsak a Fogódzóban olvasható (48. oldal) a májusi CD-mellékleten közreadott Linux Red Hat 5.2 telepítését részletesen tartalmazó leírás. A Linux és a StarOffice használatával kapcsolatos kérdésekre júliusi és augusztusi számunkban is visszatérünk, megválaszolva az olvasóink által leggyakrabban feltett kérdéseket.

Simay Endre István


Euro PLUS+ REWARD
Internet http://www.allegro.hu/reward
inter@ktív
nyelvi
kurzusok
angol
Simon Greenall
uses ViaVoice Technology
elender internet
2 hónap ingyenes
internet hozzáférés
Professional Pack
3 nyelvi szint
7 CD-n
Professional
Upgrade
4. nyelvi szint
Professional LAN Pack
Iskolai, intézményi
változatok
Euro
PLUS+
REWARD
1012 Budapest, Várfok utca 9. Tel.: 214-8621, Fax: 214-8623
E-mail: reward@allegro.hu http://www.allegro.hu/reward
ALLEGRO
INFORMÁCIÓKÉRÉS: 02

Kritikai vitriológia

Mindent bele?

E havi cikkemben néhány kedvencem újabb változatára hívom fel a figyelmet, majd két hasznos és két szép programot mutatok be, nem hallgatva el, hogy egy program tökéletessége inkább minőségi, semmint mennyiségi kérdés, még ha a programozók ellenállhatatlan vágyat is éreznek művüket az újabb változatokban minden kacattal feltupírozni.

Jurij Szalnyikov, a Real Clipboard szerzője nem lett milliomos. Ez nyilvánvaló, hiszen programjának új, 2.3-as változatában megtakarított pénzzé igyekszik konvertálni terméke áldásos szolgáltatásait. Ennyi meg ennyi centet spóroltam, figyelmeztet a program, azáltal, hogy a Real Clipboard többszörös vágólappal manipulálok. Én személy szerint szöveges makrótárolónak használom. Interneten keresztüli hitelkártyás vásárláskor az elektronikus űrlapok kitöltéséhez nyújt nagy segítséget, minimalizálja a gépelést. Az állandó adatok gombnyomásra és egérkattintásra előhúzhatók. Az új változatban immár a tálca (taskbar) jobb oldalán egy stilizált gemkapocs ikonocska is csücsül, ezzel aktivizálhatjuk a korábban csak Alt+Shift+Ins ébresztésű programot. Sőt, akinek ez nem elegendő, ballagjon el az egérkurzorral a képernyő tetejére, időzzön ott vagy két másodpercet, és a Real Clipboard akkor is ott terem. Kevésbé izgalmas változás a kezdetleges makrózási lehetőség, amellyel például egyesével felfelé lépegető számlálót lehet működtetni és behelyettesíteni a szöveges üzenetekbe. A méret még mindig kicsi, és maradt a program eleganciája is.

Juszt is Gozilla

A letöltő csodafenomén, Go!Zilla is öregedett verziószámában két tizednyit, immár v3.5. (Nevét nem vagyok hajlandó a továbbiakban a fenti helyes ortográfiával írni, szólítsa őt így az, akinek a felkiáltójel benne van a szókincsében.) Funkcionalitásában nem történt eget rengető módosulás, a program továbbra is teszi dolgát, és ez manapság nem lebecsülendő. Ha pedig valami nem lenne ínyünkre a nóvumok közül, akkor nyugodtan lépjünk át rajta, ne használjuk. A Gozilla eddig malmozgatott unalmában, mialatt arra várt, hogy az internetes böngészővel letöltendőnek kijelölt fájlra önfeláldozóan rávethesse magát. A GizmoNet fejlesztőcsapat úgy gondolhatta, hogy a holt idő alatt — mai divatos-undorító műszóval élve — „tartalmat szolgáltat". A Content Channel általuk kiválasztott shareware programokra, ingyenesen elérhető játékokra, MP3-as formátumú zenékre és egyéb mifeneségekre mutat. Magunk is készíthetünk ilyen „Ezt próbáld ki" csatornákat, ha nincs jobb dolgunk. Ezek azonban láthatóvá válnak világszerte a többi aktív Gozilla felhasználó számára is, tehát nem árt odafigyelni, nehogy lejárassuk magunkat, még ha névtelenül is.

A dizájnos új grafikus felület első látásra egy picit meghökkentő — maradjunk annyiban, hogy nem a szemléletességet fokozza.

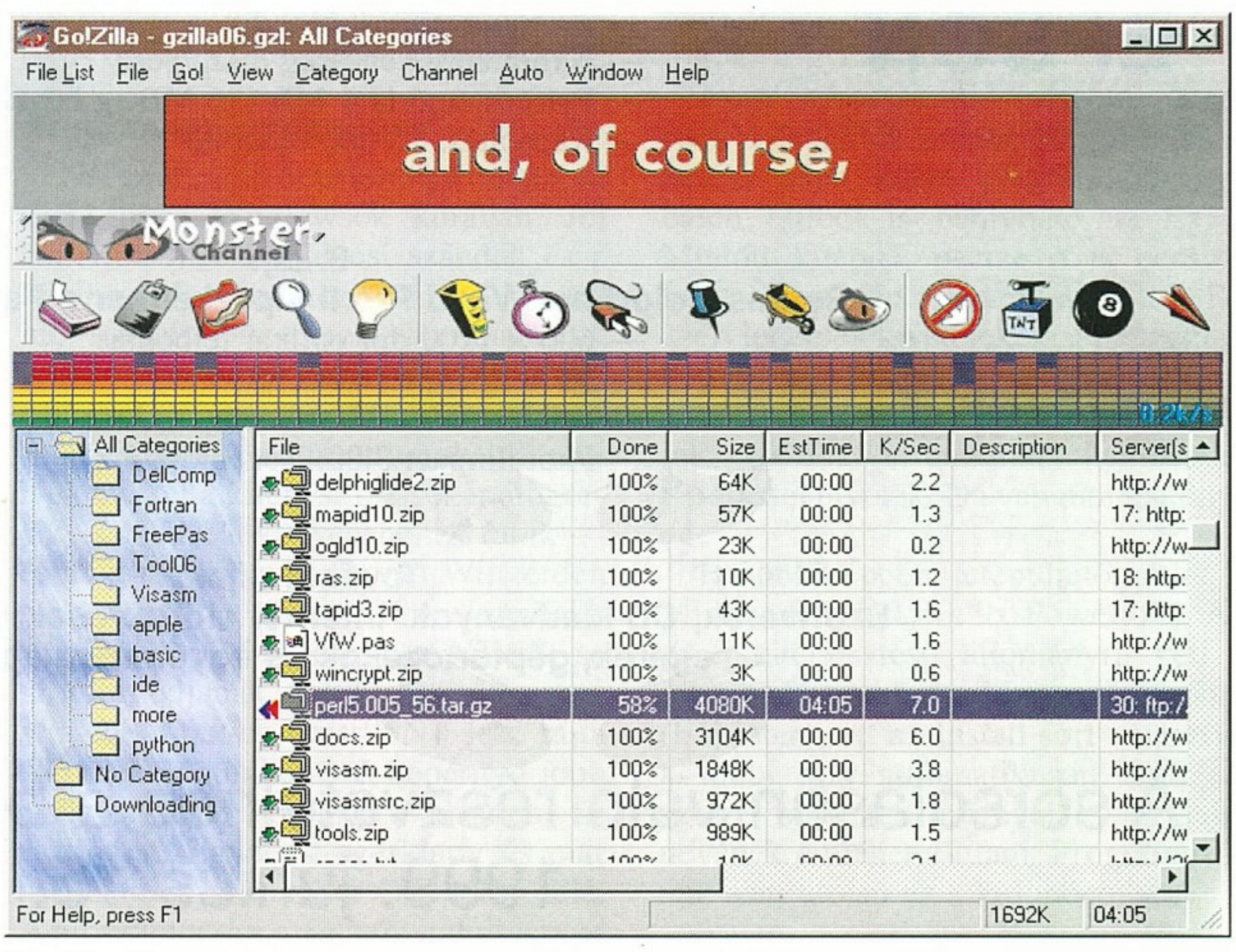
Eddig ha a Gozilla ikonjára rápottyantottuk a fájlnevet tartalmazó linket, abból láttuk, vajon képes-e a Gozilla a letöltés elkapására, hogy a kis görbe nyilacskás ikon megjelent-e vagy sem a gorillaszem megközelítésekor. Ha nem, akkor külön klatty, indulhatott a browser saját letöltő modulja. Most már erre a külön egérkattintásra nincs szükség, a Microsoft Internet Explorer 5.0 optimalizált támogatása magyarra lefordítva ugyanis annyit jelent, hogy FTP könyvtárt tükröz, vagyis egy FTP directorynak minden egyes állományát vagy az általunk megadott maszknak megfelelő fájlokat teszi magáévá a kiéhezett hímgorilla. Ha kell, az adott FTP site teljes, hierachikus könytárstruktúráját is reprodukálja.

Multi Link Leech: egyszerre több webhelyről is csipegethetünk, és a morzsák szimultán töltődnek ugyanazon bugyorba. Ha ilyenkor jó sok állományt jelölünk ki, a letöltésre várakozók sorrendje tetszés szerint cserélgethető. Megrögzött downloaderek nagy „batch"-ben fogják tartani... Legvégül immár minden egyes letöltésre kijelölt fájlnak saját, individuális Gozilla paraméterei lehetnek, amelyek menet közben is kapásból megváltoztathatók, úgy mint szerverváltás, a gorillaüvöltés engedélyzése és letiltása, a nekirugaszkodások maximális száma, az ezek közötti várakozás időtartama stb...

Hogy mit hiányolok tulajdonképpen a Gozillából? Amit megtaláltam egy másik segédprogramban. Nos itt következik az e havi legnagyobb attrakció, a Download Accelerator 2.58 (továbbiakban DA258).

Csalás, nem ámítás

A „letöltésgyorsító" mint elnevezés ugyan közhelyszerű és elcsépeltnek tű-



Mindenki nyer...

...ha kitölti és beküldi az Új Alaplap múlt havi számában elhelyezett közvéleménykutatási kérdőívet

Minden olvasó nyer, mert véleményét a lap további szerkesztése során sokkal inkább figyelembe tudjuk venni

Minden hirdető nyer, mert reklámüzenetét az olvasótáborról kapott pontos információk alapján tudja célba juttatni

Szánjon rá tehát egy kis időt, ha még nem tette volna meg

Ráadásul a legszerencsésebbek értékes díjakat nyerhetnek a sorsoláson

DELL®

Főnyeremény:
Dell OptiPlex E1 asztali számítógép a HumanSoft Kft-től

— Intel Celeron processzor, 400 MHz, 128 KB cache
— 32 MB SDRAM
— 4,3 GB merevlemez
— Magyar Win98 operációs rendszer
— 15" monitor

HUMANSOFT www.humansoft.hu

EPSON

2. díj:
Epson EPL-5700 lézernyomtató az R.A.Trade Kft-től

— Felbontás 600 × 600 dpi
— Nyomtatási sebesség 8 lap/perc
— Puffermemória 4 MB (36 MB-ig bővíthető)
— Kapacitás: 15 000 oldal/hó

TRADE R.A. Trade Kft.

Genius

3. díj:
Genius ColorPage-Vivid Pro II lapszkenner a Fan Computertől

— 600 × 1200 dpi optikai felbontás
— 9600 dpi szoftveres felbontás
— Printer portra csatlakozás
— PhotoImpact 3.02, Recognita 3.2

FAN

További díjak:
Szoftverek, CD-kiadványok, Internet előfizetések, mágneslemezek, egerek, géptartozékok és egyéb meglepetések

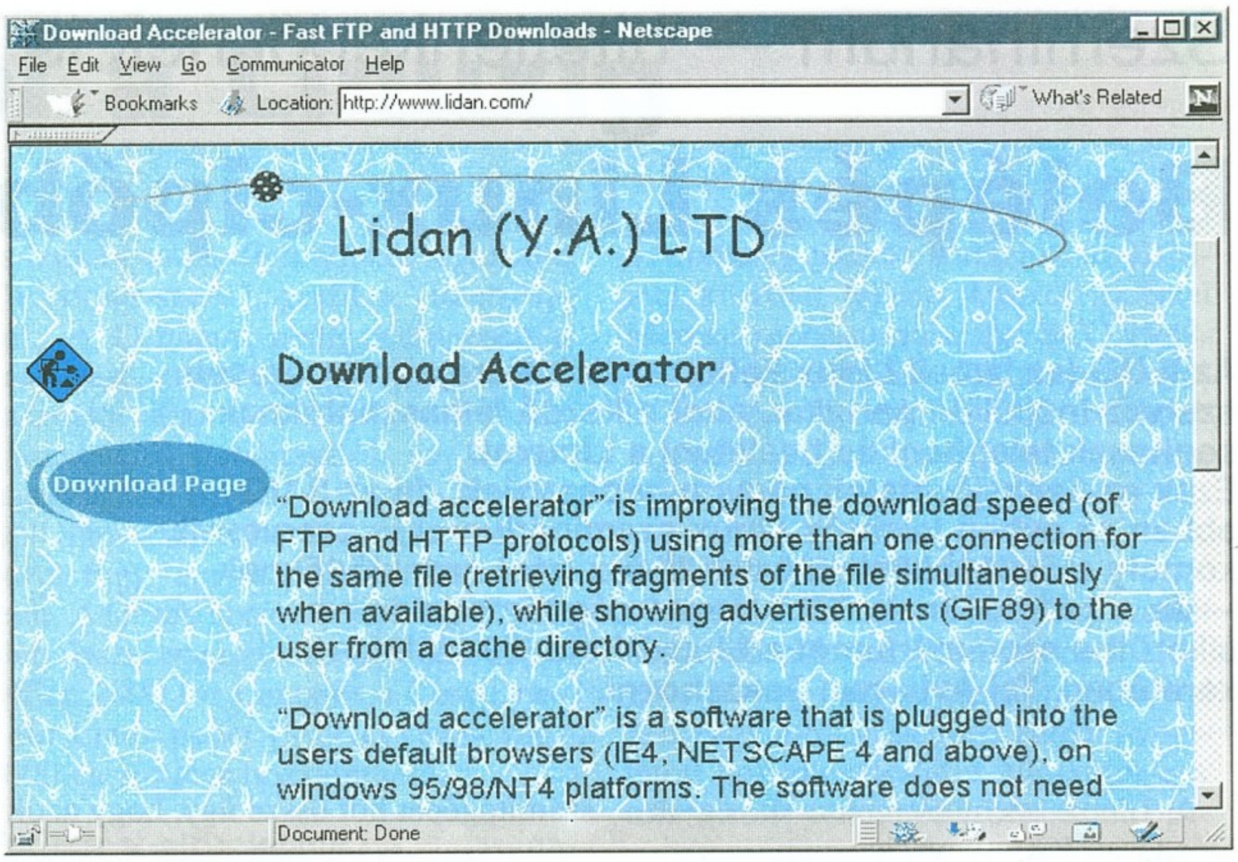
A sorsoláson való részvételhez a beküldési határidő: 1999. június 15.

nik, pedig telitalálat. A Lidan Software (www.lidan.com) spártai egyszerűségű, már-már primitívnek tetsző weblapja kevesebb sorban mutatja be ezt a zseniális programot, mint jelen sorok írója.

Hogyan szerezhetjük meg teljesen biztosan a 100 méteres síkfutás olimpiai aranyérmét? A Rejtő Jenőtől gyakran idézett megoldás mellett van még egy lehetőség: mind a nyolc sávon nekünk kell futni. A példa kicsit sántít, de gondoljunk csak a telefonvonalon lógó modemre, itt az adatátviteli sávszélesség adott, felülről korlátos. Jó esetben 56 kbit/sec, de többnyire csak 28,8. Egyetlen fix szélességű sávban kell futnunk. Most a Magyar Atlétikai Szövetség ne figyeljen. Négyszer százas váltófutás van műsoron, vagyis egy teljes kör lefutása minél gyorsabban. Normális esetben minden futó cipeli a váltóbotot és átadja az őt felváltó futónak, és egy csapat időben négy egymás utáni száz métert fut. A mi futóversenyünk kicsit más. Itt a négy (azonos) csapattag a startpisztoly eldördülésekor egyszerre indul, és közel azonos időket elérve futja le a maga távját, így a teljes kört közel negyedannyi idő alatt lehet megtenni, mint szabályosan. És még stafétabotra sincs szükség. Valahogy így működik a DA258.

Ha a letöltendő fájlt magában foglaló HTTP vagy FTP szerver maga is támogatja a fájlon belüli címzést (sajna ez nem mindig igaz, de nagy teljesítményű szerverek és webhelyek esetén már szinte biztosra mehetünk), akkor a DA258 virtuálisan több, például 6 vagy 8 részre felszeleteli a letöltendő állományt, és egyszerre kezdi el mindegyiket töltögetni, a hálózattechnikát tekintve egymástól független adatcsatornákon. Ezzel többszörösen kihasználja a szervert, amely ezt érzékelve egyre nagyobb prioritást ad a DA258-nak. Önmagát erősítő folyamat! A DA258 nélküli világban ennek éppen az ellenkezőjét tapasztalhatjuk. A letöltés folyamatosan lassul, majd amikor a túloldali szerver úgy gondolja, hogy már elaludtunk, szépen ledob minket magáról. Mi meg kezdhetjük elölről — vagy folytathatjuk Gozillánkkal a megszakadás pillanatától.

A DA258-cal tényleg nem ritka a 3-5-szörös gyorsulás. Modemen keresztül is! Ebből látszik, hogy még mindig vannak tartalékok a túlterheltnek mondott és gondolt Interneten. Visszatérve az atlétikai hasonlatra: akár szemben is futhatnak egymással. Telepítés után semmi teendőnk nincs a programmal. Ha tud, beavatkozik, amit abból látunk, hogy unaloműzésképpen



reklámokat váltogat egy keskeny sávon, a számlálót pedig a normál letöltő ablakhoz képest felpörgeti. Egyébként a letöltés megkezdésekor módunkban áll eldönteni, hogy szabályosan futjuk le a távot, vagy pedig „csalunk".

A DA258 egyelőre ingyenes. Na és milyen a DA258 együttélése a Gozillával, kérdezhetik méltán. A kérdés jó. Hát bizony, a Gozilla mohóbban kap a konc után, amit úgy tudunk elrántani előle, hogy a letöltési linkre való kattintás közben rátámaszkodunk az Alt gombra, ezzel a Gozilla farkára lépünk, és így bejöhet a DA258, ha az ellenfél szervere erre módot ad. Ha nem, akkor engedjük fel az Alt gombot, hadd csócsáljon kedvére emberszabású barátunk. A DA258 előretessékelésének másik módja az, hogy a Gozilla automatikus browser integritását kikapcsoljuk.

Angolosan távozni...

Na, ez az, amit nem könnyű megtenni a 32 bites Windowsok korában. Jól ismerjük már a kilépési szándék és a kikapcsológomb megnyomása közötti idő idegtépő üzeneteit, harddiszkünk felsikoltását és folyamatos gyötrődését, a bugyuta várakoztató feliratokat. Ismerünk viszont olyan programokat is, melyek ezeket a kilépési fázisokat átugorják. Közülük legügyesebbnek tűnik — és még karcsú is — P. van Wingerden programja, az Exits95/98. Legújabb verziója a v7.5. (Hogy az NT-sek se szomorkodjanak, nekik a némileg szerényebb v.6.x-es Exits95NT lesz rajta CD-mellékletünkön.) A képernyő jobb alsó sarkában leledzik a kiskapu, az észkijárat. Egyetlen egérklatty és már nyúlhatunk is a Power Off gomb felé. A programocska még nem indult ugyan fizikai hízásnak, de már több apró opcióval bővült. Ki-ki ítélje meg saját maga, hogy kellettek-e ezek, vagy sem. Gombnyomásra új felhasználóként való bejelentkezés (logoff). MS-DOS reboot. Screensaver azoknak, akik nem kímélik a képernyőjüket. Időzített kilépés. Billentyűkombinációra kilépés: Ctrl+Alt+X. Standby és power-off a laptop/notebook felhasználók őszinte megelégedésére.

Akiket nyomaszt ez a sok flanc, használják az előző változatot. Sajnos az egy dologra igazán jól használható programok mindig áldozatul esnek annak a jóhiszemű programozói törekvésnek, hogy a programok egyre komplexebb módon egyre több dolgot tudjanak csinálni. Aki sok bába között sokat markol (szülész főorvos), az keveset fog, és a gyerek is elvész.

Nem lebecsülendő viszont, hogy ezek a funkciók parancssorból is elérhetők, tehát külső programból vagy batch fájlból is behívható az Exits. Mielőtt kilépne, persze ő is becsuk, lezár maga után mindent, amit tud. De azért lehetnek kivételek, kísérletezzünk bátran. Egyébként is, ha oda az adat, akkor rosszak az archiválási szokásaink. Az adat is exitálhat, nemdebár? Kilépettel távozik. Ha van munkakönyve.

Hasonlóképpen telepíthető, mint a Microsoft Windows 9x Powertoys. Az .INF kiterjesztésű állományra kell a jobb egérgombbal rálőni az Intézőből (Explorer), ott az Install sort választva a többi már automatikusan lefut. A szerző ingyenes művével kíván saját nevének emléket állítani. Programozónak sem utolsó ez a kidobóember.

Herczeg József

Szeminárium — hitetlenkedőknek is

Belépés Linux-cipőben

Az ILUG (az Izraeli Linux Felhasználók Köre) rendszeresen szervez installáló partykat, ahol a kezdő felhasználóknak segítenek telepíteni és beállítani a Linuxot. Ezek az összejövetelek nagyon népszerűek, de az ILUG ebben az évben jelentős Linux szemináriumot is szervezett, komoly előadásokkal az üzleti Linux iránt érdeklődőknek — és a kétkedőknek. Több neves cég támogatásával sikerült a szemináriumot a LinuxWorld '99 kiállítás keretében, Tel Aviv-ban összehozni.

A Linux szemináriumon az egyik előadó az ICQ (www.icq.com) színeiben érdekes párhuzamot vetett fel az ICQ és Linux között. Az ICQ egyfajta kapcsolattartó program, amely jelzi, hogy használója mikor van a Neten, képes üzenetet, címeket, miegyebeket küldeni, és rajta keresztül beszélgetni is lehet egy hasonló programot használóval. Az ICQ-nak nem volt hagyományos értelemben vett reklámja, mégis rendkívül gyorsan elterjedt és népszerű lett. Az ICQ-t eredetileg egy Tel Aviv-i csoport készítette, kb. 3 évvel ezelőtt, és az alapításától számított rövid időn belül meg is vette az America Online, pár száz millió dollárért, ami jelzi, hogy a program minden szempontból befutott. A Linux hasonló úton jár: nincs reklámköltsége, ma is nagyrészt barátok, ismerősök útján terjed és szintén sikeres. Mindkét trend előre mutatja, hogy a modern piac gyökeresen megváltozott, és hogy „ingyenes programot írni" nem jelenti feltétlenül azt, hogy „ingyen dolgozni".

Mosix

A Jeruzsálemi Héber Egyetem (mosix.cs.huji.ac.il) színeiben Amnon Barak az általa és diákjai által kifejlesztett Mosixról szólt, és a fejlesztést lehetővé tevő, saját készítésű segédprogramokról. Előadása után hetekig nagy visszhangja volt a Mosix rendszernek, azért is, mert annak forrását nem tették közzé. A Linux kernel GPL licence bizonyos esetekben kötelezővé teszi az alatta fejlesztett programok forráskódjának közzététélét, és valójában a Mosix is ide sorolható.

A Mosix sokévi fejlesztés eredményeként előzőleg szintén egy Unix változat, a BSD alatt futott, újabban azonban Linuxra fejlesztik. A program több számítógépet köt össze úgy, hogy azok erőforrásai, számítókapacitásai összeadódnak, és a kivülálló számára, aki ezt a rendszert használja, az egész egyetlen nagy és meglehetősen gyors számítógépnek látszik. Az egyetem 150 darab (Pentium és PII-es processzorú) számítógépből álló hálózatot kapcsolt össze, Gigabit Ethernettel, így az Izrael leggyorsabb, egy egységként működő számítógépe lett. A gépen számolásigényes programokat, például törési, ütközési vizsgálatokat, biológiai szimulációkat végeznek. Az előadáson hallott érdekes információ volt, hogy az egyetem megkapta a Microsofttól az NT forráskódját, hogy alakítsák azt át a Mosix rendszerrel történő használatra, de az NT kódjának megtekintése után megállapították, hogy az teljességgel lehetetlen, ezért az ötletet elvetették.

Patinás nevek

A Mosix jelenleg csak maximum 6 gépes klaszteres verzióban hozzáférhető az Interneten. Ez a korlátozás azért sajnálatos, mert a Mosix jelenleg a világon egyedülállóan olvasztja egybe több gép erőforrásait. A nagyobb teljesítményű változatok visszatartása tehát lefékezi azt a folyamatot, melynek eredményeként előbb-utóbb bárki rendelkezhet majd könnyen kezelhető hatalmas számítókapacitással. (A másik hasonló kezdeményezésű projekt, a Beowulf nyílt forráskódú.) A Mosix project keretében összekapcsolt gépeken lévő Linux kernel mindig a legfrissebb verzió. A Mosix program forrásának kiadása körül ugyan még tart a huzavona, miután azonban költséges munka volt, a finanszírozó szponzor szava a döntő, és az egyelőre nemleges.

A Compaq álláspontját a cég egyik izraeli vezetője ismertette. A Compaq nyitott az újítások felé, de csak jól működő rendszereket adnak ki a kezükből. Ha valaki Linux operációs rendszerű gépet rendel a cégtől, a rendszert a helyi Linux disztributor installálja és támogatja. Az így felszerelt gépen nem használják fel a Linux minden újdonságát, csak a már régóta megbízhatóan működő programokat, modulokat teszik fel. A Compaq feladata úgy összerakni gépeit a Linux alá, hogy azok „bombabiztosak" legyenek, megfeleljenek egy szerver követelményeinek.

Az Oracle neve számomra az adatbáziskezelés szinonimája. Nos, az Oracle nemrég adta ki az Oracle 8i nevű adatbáziskezelő linuxos változatát. Ebbe a programba mindent beleraktak, ami csak kapcsolatba kerülhet az adatbáziskezeléssel és az Internettel. Egy ilyen robusztus programnak szilárd alapokon kell állnia, és ha az Oracle úgy gondolta, hogy a Linux elég szilárd alap programjának futtatására, akkor az már jelent valamit.

Fontos tanulság

Izrael legnagyobb független szoftverforgalmazója, a PF1, Linux disztribútorként mutatkozott be. Tőlük lehet rendelni Red Hat Linuxot, és támogatják a hivatalosan megvásárolt többi Linux disztribuciót is. Ebben az az érdekes, hogy fő profiljuk nem a Linux, mégis ők lettek a legnagyobb linuxos megoldásszállítók az országban.

A meghívott előadók után az egyik szervező tartott előadást a Linux héber nyelvű változatáról. Megnézhettük a Netscape héber változatát is. A program átállítható az Izraeli hálózaton használt 4 (!) különböző szabvány szerinti megjelenítésre. Néha a böngészőnek kell megfordítnaia a szavakat, néha régi DOS szerinti kódkészletet használnak, ez ügyben tehát itt a mai napig is nagy a keveredés az Interneten.

Bemutatásra került továbbá a KDE, a Linux szép és kellemes grafikus kezelői felülete, amely jelentősen megkönnyíti a Linux használatát.

A szeminárium legfőbb tanulsága, hogy a Linuxot már a nagy cégek sem hagyják figyelmen kívül, és a fejődés egyik lehetséges elágazásának tekintik.

Gervai Tamás

Windowstól a Linuxig

Linux Rulez

Út a „hivatalosodáshoz": Linux Certificate

A Linuxot forgalmazó IBM-től a Linux disztribúciót gyártó Caldera Systems cégig bezárólag egyre többen sorakoznak fel egyfajta „Linux Certificate" bevezetése mellett. Ha a tervek megvalósulnak, a jövőben a KeyLabs mint független szereplő által kiállított bizonyítványra lesz szükség ahhoz, hogy valaki Linuxot forgalmazzon. Így remélhetőleg elkerülhető egy olyan helyzet kialakulása, amelyben az egymással nem teljesen kompatibilis Linux verziók miatt széttagolódik a piac. A KeyLabs jelenleg a Java Certificate programmal foglalkozik, és a tervek szerint hamarosan olyan programokat fog a szoftverkereskedők rendelkezésére bocsátani, amelyek segítségével eldönthető, hogy az adott Linux verzió eleget tesz-e a követelményeknek.

Mivel a Linux szabad forráskódú, elvileg semmi akadálya annak, hogy valaki a bizonyítvány követelményeinek nem megfelelő Linuxot fejlesszen, a gyakorlatban azonban a nagy forgalmazók (mint például az IBM, a Dell) meg fogják követelni a KeyLabs bizonyítványát. Számos Linux-forgalmazó egyébként már ma is támogatja a Linux Standards Base előírást, amely az egységesítés érdekében meghatározza, hogy a fejlesztők milyen könyvtárakat (libraries) használhatnak, eközben pedig a Linux Professional Institute a Novell „Certified Novell Engineer" vagy a Microsoft „Certified Solutions Engineer" minősítéshez hasonlót kínál. Szabad forráskód ide vagy oda, a Linux nagyon gyorsan ugyanolyan üzleti törvényszerűségek közepette találhatja magát, mint a Microsoft. *(ZDNN)*

Linuxrakéta: Corel + KDE + Debian

A Corel két jelentős open source fejlesztő, a KDE („Kommon" Desktop Environment) és a Debian erőinek bevonásával közösen fejleszt az eddigieknél sokkal inkább felhasználóbarát installációs és grafikus felületet a Linuxhoz. A Corel azért választotta a KDE-t, mert jelenleg ez tűnik a legkönnyebben és legjobban konfigurálható, leginkább „Windows-szerű" grafikus interfésznek. A Debianra pedig azért esett a választása, mert egyrészt nagy a felhasználói tábora, másrészt ez a Linux disztribúció stabilitásáról és biztonságosságáról ismert. A Corel a Cygnus GNUPro szoftvertechnológia segítségével fogja applikációit Linuxra átültetni. A tervek szerint az új termék már idén ősszel megjelenik, a WordPerfect Office 2000 for Linux és a CorelDraw 9 for Linux pedig 1999 legvégén vagy 2000 elején lesz hozzáférhető.

Terjedési előrejelzés

Az International Data Corporation (IDC) legutóbbi felmérése szerint a Linux népszerűsége napról napra nő, és 2003-ig a Linux üzleti alkalmazásai gyorsabban fognak terjedni, mint az IDC által figyelemmel kísért más kliens vagy szerver környezet programjai. Az IDC 1999 és 2003 között évi 25%-os növekedésre számít, miközben ez a többi kliens operációs rendszernél 10%, a szerver oldalon pedig 12% lesz. *(LinuxToday)*

Torvalds kedvencei: a RedHat és a SuSe

Linus Torvalds egy interjúban elmondta, hogy bár a Linux disztribúciói meglehetősen hasonlítanak egymásra, neki a Red Hat és a SuSe a két kedvence. Szerinte a jelenleg 2.2-es verziónál tartó Linux kernel ebben az évben egyre több olyan vonást tud felvonultatni, amely elősegíti az ipari méretű, „high end" alkalmazást. Bár örült annak, hogy a Linux mint a Microsoft Windows lehetséges riválisa került reflektorfénybe, arra nem számít, hogy a Linux valaha is az első számú, monopol helyzetű operációs rendszer lesz. *(ZDNet)*

Tisztességes-e a Red Hat?

Doug Michels, a Unix egyik legnagyobb forgalmazójának, az 1979-ben alapított SCO-nak a vezetője azt nyilatkozta, hogy miközben a Linux modelljének feltétlen híve, tisztességtelennek tartja a Red Hat eljárását, mert nagy üzletet csinál a szabad forráskódból, vagyis tulajdonképpen mások önzetlenül végzett munkájának eredményéből húz hasznot. Az SCO társalapítója elmondta, hogy cége jelenleg például Unixot ad el kisvállakozásoknak, illetve a Monterey projekt keretében az IBM-mel közösen fejleszt Unixot az Intel Merced processzorára. Az SCO 1998-ban 55 millió dolláros forgalmat bonyolított le, és eddig mintegy 200 ezer darab ingyenes SCO Unixot osztott szét nem üzleti célú használatra. *(Newswire)*

A Linux üzletiesedése

Az idei tavaszi Comdexen tartott beszédében Linus Torvalds nem kifogásolta, hogy olyan cégek, mint a Red Hat vagy a SuSe pénzért adnak nyomtatott dokumentációt és supportot egy tulajdonképpen szabad forráskódú operációs rendszer mellé. „Minden kereskedelmi célú befektetés hozzásegíti a Linuxot ahhoz, hogy kiegyensúlyozott rendszerré váljon és még inkább elterjedjen". A piacorientált cégek közreműködése nélkül ugyanis a Linux nem tudna kilépni a számítógépes közösségből, nem válhatna széles körben használt, felhasználóbarát operációs rendszerré. Bill Gates továbbra sem hiszi, hogy neki a szabad forráskódú operációs rendszer komoly kihívást jelent. Szerinte legfeljebb szövegszerkesztőket és táblázatkezelőket fognak rajta széles körben futtatni, de a felhasználók nagy része megmarad a Windows mellett, annak „egységesebb szolgáltatásai" miatt. *(Chicago Tribune)*

Az AOL közeledik a Linuxhoz

Az America Online alig egy éve főleg Internet-szolgáltató volt. Azután felvásárolta a Netscape-et, és most a jelek szerint további nagy tervei vannak, melyekhez a Windows helyett a Java és a Linux jöhet számításba. Az AOL a Sun-nal együttműködve kidolgozta az AOL Anywhere programot, amellyel a felhasználók nem csupán személyi számítógépeken keresztül érhetik el az Internetet. Emellett Linux alapú olcsó számítógépek tennék lehetővé a kisebb pénzű felhasználóknak is a hozzáférést. Könnyen elképzelhető, hogy az AOL a hardverpiacra is be kíván lépni. *(CNET)*

Itt a Red Hat 6.0!

Az 1998-ban megjelent Red Hat 5.2 a legelismertebb és az üzleti életben is legtöbbször használt Linux disztribúcióvá vált. A most elkészült Red Hat 6.0 valószínűleg tovább folytatja a hódítást. Ebbe már bele van építve többek között az IBM ViaVoice hangfelismerő

rendszer, a Dell pedig előinstallálva adja szervereihez, mainframe gépeihez és a PC-khez is. A szoftver kereskedelmi ára 40 és 80 dollár között mozog, de a 6.0-s szabad verzió letölthető a Red Hat weblapjáról. *(LinuxWorld)*

Ingyen PC után: ingyen szerver...

A tavaszi Internet World-ön a Novell korábbi vezetője, Bob Frankenberg bejelentette, hogy az általa alapított Encanto Networks (http://:www.encanto.com/products/freeego/frameset.html) ingyen fogja adni e.go nevű webszerver gépét, amennyiben a „vásárló" hajlandó a céggel két évre szóló, havi 69 dolláros szerződést kötni. Cserébe egy Cyrix MII-vel és 32 MB RAM-mal felszerelt gépet kap, 2 GB-os merevlemezzel és négyportos Ethernet kártyával, valamint egy 56 kbit/sec-os modemet, illetve a BSD Unix egyik verzióját, melyet látványos, browser jellegű felületen lehet konfigurálni. A szoftver webszerver és emailszerver funkciót tölt be, valamint lehetővé teszi website építését és az elektronikus kereskedelemet. Az ár magában foglalja az Encanto cég InstantConnect Service szolgáltatását, ezáltal a cégen keresztül történő hitelkártya-tranzakciók mellett az Encanto dedikált proxy/cache szervereinek használatát is. Vagyis az e.go szerverek weblapjai a cache-ben vannak elhelyezve, ezáltal jóval gyorsabb a kommunikáció, mintha mindig az 56 k-s modemre kellene hagyatkozni.

...és olcsó szerver

Az Internet World-ön aktivizálódtak az olcsó szervermegoldásokat kínáló cégek. A Cobalt Network (http://:www .cobaltnet.com/products/index.html) Linux alapú Qube2 és QubeRaq 2 megoldásokkal állt elő, melyek ugyan nem ingyenesek, de kevesebb, mint 1000 dollárért robusztus intranet/Internet/e-mail szervert, valamint 250 MHz MIPS-es processzorral felszerelt gépet kínálnak. Az installálás (akárcsak az e.go esetében) a kicsomagolástól számítva kevesebb, mint 15 percet vesz igénybe (a szerverek körében ez eddig nem nagyon volt jellemző). Ráadásul a Qube html alapú adminisztrációja a hálózatra kapcsolt bármelyik PC-ről lebonyolítható. A kisebb vállalkozások számára ezek az új típusú, olcsó és könnyen kezelhető szerverek ideális megoldást jelentenek. *(InfoWorld)*

A Mozilla bukásának tanulságai

A Mozilla egyik guruja, Jamie Zawinski otthagyta az AOL-t, mert szerinte a szabad forráskódon alapuló böngésző kudarcnak bizonyult. Zawinski úgy véli, hogy a Netscape már 1998 elején elvesztette a browserháborút a Microsoft Internet Explorere ellen, ráadásul képtelen volt megfelelő érdeklődést kelteni a Netscape szabad kódján alapuló Mozilla iránt, amelyen emiatt kb. 100 fős Netscape programozó csapatnak kellett dolgoznia. Ráadásul egy év alatt sem sikerült a Mozilla béta-verzióját befejezni. Szinte teljesen át kellett írni az egész kódot, hogy a Gecko kódnevű „kirajzoló motorra" (rendering engine) lehessen ráépíteni az új browsert, és újra kellett írni a felhasználói felületet is. Végkövetkeztetése, hogy a szabad forráskód sem olyan egyetemes csodagyógyszer, amely megoldást kínál bármilyen problémára. *(Wired News)*

Sinclair és a Linux

Lord Sinclair, a ZX-81 és a Sinclair Spectrum számítógépek megalkotója az informatikai iparba való visszatérést fontolgatja a PC-nél olcsóbb, Linux operációs rendszerű számítógép megépítésével. Sinclair bízik abban, hogy ötlete támogatásra talál, hiszen a Linux mind népszerűbbé válik az üzleti szférában is. Véleménye szerint a PC-k ára azért magas, mert az Intel drágán kínálja processzorait, a szoftverek memóriaigénye nagy, és maguk a szoftverek is indokolatlanul sokba kerülnek. Az új Sinclair gép olcsóbb processzort és egyszerűbb tápegységet tartalmazna, kevesebb memóriát igényelne és szabad operációs rendszerrel (Linuxszal) működne. A konstruktőr szerint az új számítógép akár feleannyiba is kerülhet, mint most egy átlagos PC. *(news desk)*

Csúcstalálkozó a jövőről

A legnagyobb Linux-terjesztő cégek több vezetője (például a Red Hat-től Bob Young, a Caldera-tól Ransom Love, a VA Research-től Larry Augustin) mintegy 250 résztvevő előtt folytatott beszélgetést a tavaszi Comdex alatt a Linux jövőjéről. Augustin szerint „a Linux még ebben az évben maga mögé utasítja a Windows NT-t, és a legelterjedtebb operációs rendszer lesz a szervergépeken". A Linux az NC kategóriában is tarolni fog, ám a szoftverfejlesztők jelentős része egyelőre még ódzkódik tőle, mert attól tart, hogy a Linux nem elég egységes. Ezzel kapcsolatban többen is támadták a Red Hat-et, mert a jelek szerint saját verzióját akarja szabvánnyá tenni, Young azonban kijelentette, hogy együtt kíván működni a többiekkel. Ugyanakkor a Linux PPC, amely Macintoshra szállít Linuxot, azt tervezi, hogy a jövőben a szerver oldal helyett a desktopra fogja helyezni a hangsúlyt. „Kezdőknek még mindig nem olyan egyszerű a Linuxot használni, mint a Macintosht vagy a Windowst", állapította meg Jeff Carr, a Linux PPC alapítója. Augustin arról számolt be, hogy üzleti hasznának jelenleg több mint 60%-a származik a nem hobby, hanem üzleti célú Linux felhasználóktól, a Calderánál ez az arány a fél évvel korábbi 67%-ról már 85-90%-ra ment fel. *(ZDNet)*

Windose

Újabb (vitatott) Mindcraft teszt

A Mindcraft tesztje szerint a Microsoft Windows NT gyorsabb fájlszerver és webszerver, mint a Linux. A tesztet a Microsoft fizette, ő szállította a gépeket, ő biztosította a labort — és ő nyert. Érdekes módon a ZDNet és a Mindcraft majdnem egyidejűleg végzett közös tesztje ennek éppen az ellenkezőjét mutatta ki. A Mindcraft végezte korábban a NetWare és az NT hálózatok összehasonlítását is, amelynek során szintén az NT bizonyult jobbnak, és az eredmény jókora botrányt kavart, mert a Novell állítása szerint az NT hálózatot optimalizálták, a Novell hálózaton pedig még azokat a beállításokat sem végezték el, amelyeket a rendszerhez mellékelt kézikönyv javasolt. Sokan a Microsoft és a Mindcraft közös érdekeltségeit kezdték firtatni, és megkérdőjelezték a tesztelések pártatlanságát. Most is botrány van kibontakozóban. A ZDNet az eredményeket illetően kételkedésének adott hangot, Linus Torvalds pedig azt állítja hogy megkeresték ugyan őket, de javaslataikat egyszerűen nem vették figyelembe. A teszt során Red Hat Linuxot használtak, amelyen a telepítés után további beállításokat nem végeztek. A teszt előtt érkezett egy felkérés a Mindcrafttól egy linuxos hírcsoportba is, hogy segítsenek a Linux konfigurálásában, mely kérés azonban a LinuxToday szerint a Microsoft hálózatból (tide71.microsoft.com) a Dejanews szerveren keresztül érkezett, ami véleményük szerint a Mindcraft és a Microsoft közös érdekeltségét jelzi. (Részletek: http://linuxtoday.com/stories/5323.html). A kérésre a linuxosok természetesen még aznap válaszoltak, de visszajelést nem kaptak.

A Mindcraft ígéretet tett a tesztek újbóli elvégzésére, ezúttal a linuxos tábor által ajánlott beállítási paraméterekkel, de a Red Hat szakembereit vagy Linus Torvaldst nem állt szándékukban a tesztlaborba beengedni, viszont a friss hírek szerint ezt mégis lehetővé teszik. Torvalds úgy nyilatkozott, hogy nem sok jó várható az olyan teszteléstől, amely nem megfelelő körülmények között zajlik. Az általa látott tesztekben a Linux egyszerűen „lemosta a pályáról" az NT-t. *(ABCnews, LinuxToday)*

A Windows szakít a Windows-zal

A megbízhatóság és a stabilitás érdekében a Microsoft már nem vállalja, hogy a Windows 2000 kompatibilis legyen visszafelé a korábbi Windows verziókkal — jelentette be Jim Allchin, a Microsoft alelnöke, aki egyben a Windows 2000 munkacsoport vezetője is. Ettől a lépéstől a Microsoft vezetői azt remélik, hogy az új operációs rendszert megszabadíthatják a gyakori lefagyások miatt ráragasztott „kék halál" elnevezéstől. Jim Allchin szerint az instabilitás okai főként a nem Microsoft által készített driverek, virtuális eszközmeghajtók és DLL-ek. Az okok másik csoportját az ISA kártyákat kezelő rezidens és nem rezidens programok alkotják — az alelnök szerint ezekből nem lehet stabil rendszert építeni.

Mindez azt jelenti, hogy azok, akik a Win2000 szerver telepítése mellett döntenek, kénytelenek lesznek a kliens oldalon működő Win 3.x, Win95, Win98 rendszereiket is Windows 2000 Pro-ra cserélni. Ez automatikusan maga után vonja a hardverek tömeges kényszerű cseréjét, az új Windows ugyanis egyrészt sokkal erőforrásigényesebb, másrészt a korábbi hardverelemek közül nem mindegyiket tudja kezelni. (A korábbi Windows alkalmazások közül sem mindegyik fut az új rendszeren.)

A Microsoft szokatlanul hosszú ideig ragaszkodott a „visszafelé kompatibilitás" elvéhez, és sok problémának többek között ez a törekvés a forrása. A korábbi Windows változatoktól való teljes elszakadásnak egyenes következménye viszont az, hogy a Microsoft mégis kénytelen az egyszer már lezártnak tekintett Win95/98 termékvonalat is fejleszteni: túl sokan vannak ugyanis, akik nem vennék jó néven, hogy a korábban drága pénzen megvásárolt programok egyszeriben hasznavehetetlenné válnak az új Windows operációs rendszeren.

A Windows 2000 harmadik béta-verzióját a korábban jelzett április végi időponttól eltérően csak májusban adják át tesztelésre mintegy félmillió felhasználónak. A hivatalos álláspont szerint a végleges változat a csúszások ellenére még ebben az évben megjelenik. *(Computerworld)*

További dátumkezelési hibák

A Microsoft közleménye szerint három további dátumkezelési hibát fedeztek fel a Windows 98 operációs rendszerben. Az elsőnek a lelőhelye a Microsoft Automation Library, amely 2029 után minden dátumot 20. századinak értelmez, például 2030-at 1930-nak. A második hiba a nemzetközi dátumformátumot érinti (év-hónap-nap), amely az évek rögzítésére két számjegyet használ, és a 00-át nem 2000-nek értelmezi például a DOS-os xcopy parancs használatakor. A harmadik a Win98 nemzetközi változatában található, és akkor jelentkezik, amikor valaki Visual Basic-et használó alkalmazásokat futtat. Ekkor azok ugyanis az alapnaptárat (default calendar) használják, függetlenül attól, hogy a felhasználó melyiket jelölte ki. David Weeks, a Windows marketing menedzsere szerint, mivel a Win98-at otthoni felhasználóknak szánták, nem valószínű, hogy többségük valaha is találkozik a fenti hibákkal, ugyanakkor beismerte, hogy nem tudják, vannak-e további dátumkezelési hibák is a Windows 98-ban.

A hibajavító csomagok rövidesen megtalálhatók lesznek a Windows Update oldalakon, valamint a Year 2000 Resource, Y2K hibákat javító csomagokat tartalmazó CD-n. A Win98 második kiadása már mentes lesz a fent említett hibáktól. További részletek a http://www.microsoft.com/year2000 címen olvashatók. *(Infoworld, news desk)*

Testvéri konfliktus: Explorer kontra Outlook

Több internetes hírcsoportban megjelent levél szerint a Microsoft Internet Explorer 5.0 böngészője összeférhetetlen az Outlook 98 levelezőrendszerrel. Azok, akik telepítették az Explorer 5.0-t, legnagyobb meglepetésükre nem tudták telepíteni vagy újratelepíteni az Outlook 98-at, így e-mail hozzáférés nélkül maradtak. Egy terméktámogatással foglalkozó, magát megnevezni nem kívánó mérnök, aki nemzetközi nagyvállalat leányvállatánál dolgozik, közölte, hogy Európában mintegy háromezer felhasználójuknak akarták szétküldeni az új Explorert, de erről lemondtak, miután egy teszt során a fenti hiba kiderült. A mérnök elmondta, hogy azért nincs szerződésük a Microsofttal terméktámogatásra, mert cégek számára az túl drága. Így csak abban reménykedhetnek, hogy a Weben hamarosan megjelenik a hibajavító csomag. Neil Laver A Microsoft Internet termékmenedzsere nem tud a hibáról, de ha az mégis valós, akkor elkészítik és közzéteszik annak javítását. Hozzátette, hogy a rövidesen megjelenő Office 2000 magában foglalja az IE 5.0-t és az Outlook új verzióját is, ami bizonyosan megoldja a problémát. *(news desk)*

Felszabadul-e a Windows forráskódja?


CNET News.com - How far will Microsoft go with open source? - Netscape

File Edit View Go Communicator Help

Bookmarks Netsite: ews.com/News/Item/0%2C4%2C34876%2C00.html?dd.ne.

CNET : News : Enterprise Computing : Story

Front door | Enterprise Computing | E-Commerce | Communications | The Net | Personal Technology | Services and Consulting | Year 2000 | CNET Investor | CNET Radio | Perspectives | Rumor Mill | One Week View

Enterprise Computing

How far will Microsoft go with open source?

By Stephen Shankland and Stephanie Miles
Staff Writers, CNET News.com
April 9, 1999, 8:25 a.m. PT

High-level Microsoft executives twice this week have floated the possibility of making Windows 2000 software an open source project, but observers aren't buying it.

Such a move would be contrary to Microsoft's core philosophy. And it would mean the open source community would have to make room for a company that's a lightning rod for the many in the movement.

"The company understands that there are certain benefits to the open source model, but fundamentally Microsoft is a company based on the notion of intellectual property; and open source and intellectual property butt heads in the real world," said Dwight Davis, a Microsoft analyst at

http://www.news.com/cgi-bin/acc_clickthru?clickid=0000140fee7

Steve Ballmer a Windows Hardware Engineering című konferencián arról beszélt, hogy a Microsoftnak rövidesen döntenie kell forráskódjainak szabaddá tételéről. Elemzők szerint a Microsoft védekezése az ellene folyó perben meglehetősen gyenge lábakon áll, és egyre nagyobb nyomással nehezedik rá a Linux léte is, ezért könnyen lehet, hogy a pervesztés elkerülésére a Windows forráskódját mégis közlik. Ugyanakkor a megfigyelők arra is figyelmeztetnek, hogy a Windowsra fejlesztők (a felhasználók nagy részének támogatásával) a forráskód mostani kezelési módját tartják jónak, mert úgy vélik, hogy ez jobban biztosítja a technikai támogatást és a fejlesztést. Bármennyi is a panasz, tény, hogy a Windows rendszerek és a windowsos alkalmazások elterjedtségük miatt még jó darabig sok helyen jelen lesznek. Az is igaz viszont, hogy megtörténtek a nyitás első lépései, idén márciusban a Microsoft Internet Financial Toolkit (online banki szolgáltatásokat nyújtó program) forráskódját már hozáférhetővé tették. *(news desk)*

Galántai Zoltán – Mákos András

Nem csak OS/2 ...

Lotus Notes 5 kliens OS/2-re?

A német Ct magazin híradása szerint a német pénzügyi szektor azt kérte az IBM-től, hogy készítse el a Lotus Notes 5-ös kliens változatát OS/2-re is. Mint ismeretes, az IBM csak a Lotus Notes 5-ös szerverprogramját kívánta kifejleszteni OS/2-re, ám lehet, hogy a németek kérésére mégis meggondolja magát. A Lotus európai menedzsere alátámasztotta, hogy az IBM komolyan mérlegeli az OS/2-es kliens elkészítését. A németül tudók elolvashatják a sztorit a http://www.heise.de/newsticker/data/jk-21.03.99-001/ oldalon.

HomePage Publisher 2.1B

OS/2-es körökben nagyon sokan az első számú HTML szerkesztőnek tartják a HomePage Publishert, amelynek most jelent meg a 2.1B változata. A WYSIWYG (azt kapod, amit a képernyőn látsz) elvet követő alkalmazás ezen kiadásában nagymértékben tökéletesítették a Smart Publishing (intelligens publikálás) funkciót. A termékről további információ a http://ourworld.compuserve.com/homepages/clerin/hppframe.htm oldalon.

PdfBuilder/2

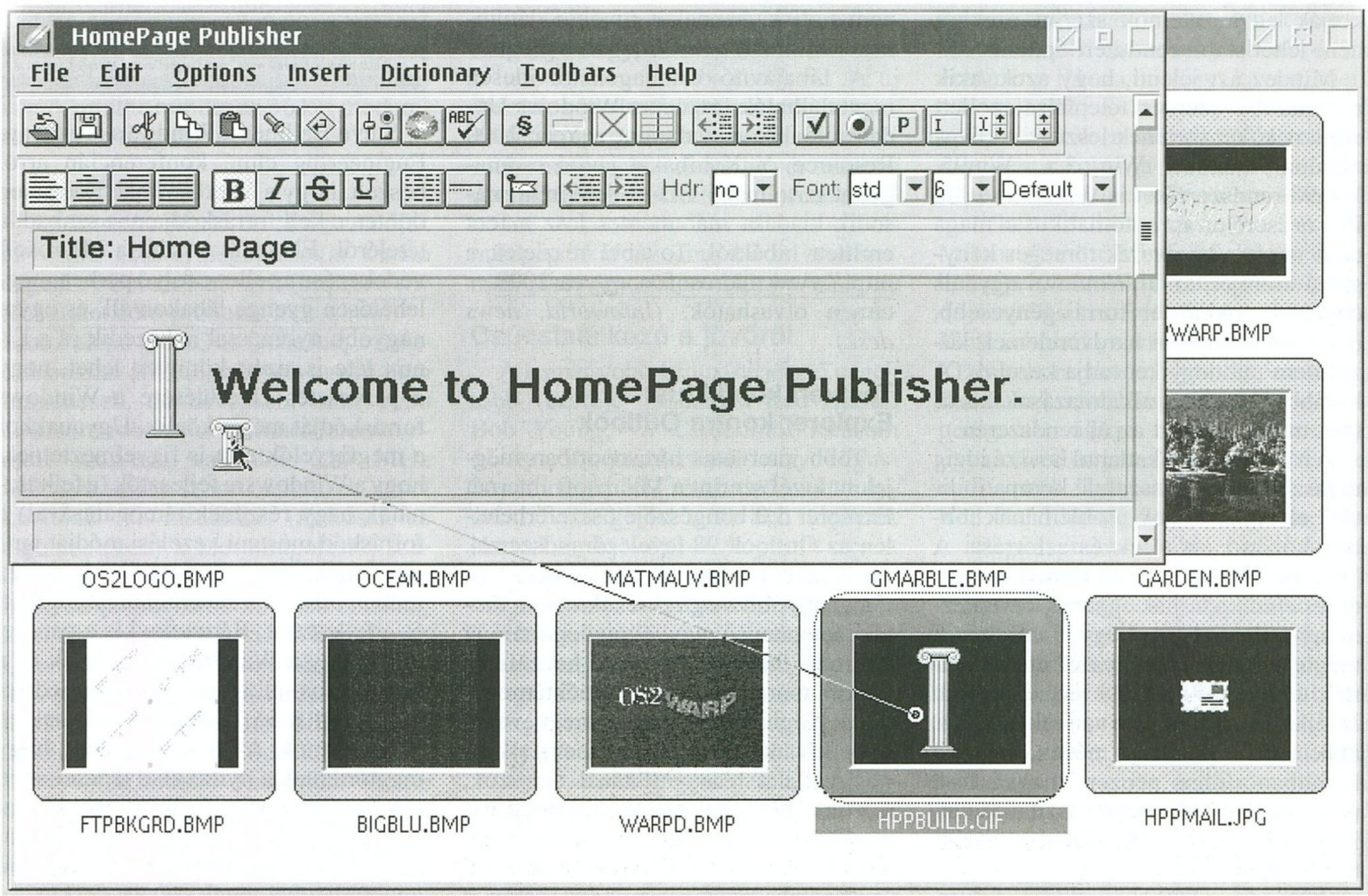
HTML fájlok PDF formátumúra alakítását végzi el a dán Dan Holmes által kifejlesztett shareware alkalmazás, a PdfBuilder/2. A jelenleg béta-állapotú program (pdfbldr1.zip) letölthető a http://hobbes.nmsu.edu-ról. A felhasználók által tanúsított nagy érdeklődés miatt az alkalmazás fejlesztője egy levelezési listát is indított, amely a http://www.egroups.com/list/pdfbuilder/ címen olvasható.

HttpTunnel/2

Elsősorban tűzfal mögött dolgozók érdeklődésére számíthat a Unix alól OS/2-re átírt HttpTunnel program, mert segítségével az Internet olyan szolgáltatásait (telnet, ftp) is igénybe lehet venni, amelyek nem minden esetben elérhetők a tűzfal mögött. Az alkalmazás a tűzfal által átengedett http protokoll felhasználásával kétirányú virtuális kapcsolatot létesít más kiszolgálókkal, és így más protokollok használatáról is szó lehet. Az alkalmazás megtalálható a http://www.viaduk.net/freepage.nsf/page/OlDer/$file/ht2_20.zip címen. Az eredeti Unix program forráskódjának fellelési helye: http://www.nocrew.org/software/httptunnel.html.

Styler/2

A korábban Smart Windows néven ismert shareware alkalmazás Styler/2 név alatt él tovább. A Smart Windows 0.82 frissítésének megfelelő első Styler/2 bővíti az OS/2-es munkaasztal funkcionalitását, valamint javít annak kinézetén is. A Styler/2-ben a fejlesztő Alessandro Cantatore kijavította a Smart Windows finom ablakmozgatás



Fogd-és-vidd módszerrel készül a honlap

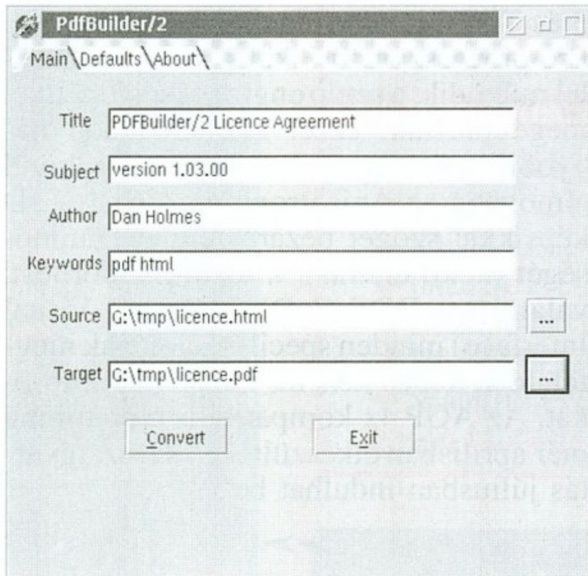
funkciójában felfedezett hibákat is. A program 90 napos próbaverziója letölthető az ftp://ftp.ghostbbs.cx/pub/os2/os2team/alcant/smw082b.zip címről.

VNC szerver

A VNC klienst OS/2-re átíró japán programozó, Akira Hatakeyama elkészítette a VNC szerver OS/2-es változatát is. A futtatáshoz egyelőre még szükség van az XFree86-ra is, így a kliensek csak az OS/2 alatt futtatott X munkaasztalt távirányíthatják. A szerver kibocsátása előtt néhány nappal a PM VNC kliens is frissült, és a cikk írásának pillanatában már a 6-os bétánál tartott. A programok letölthetők a http://www.sra.co.jp/people/akira/os2/index-e.html oldalról kiindulva.

Javítások minden mennyiségben

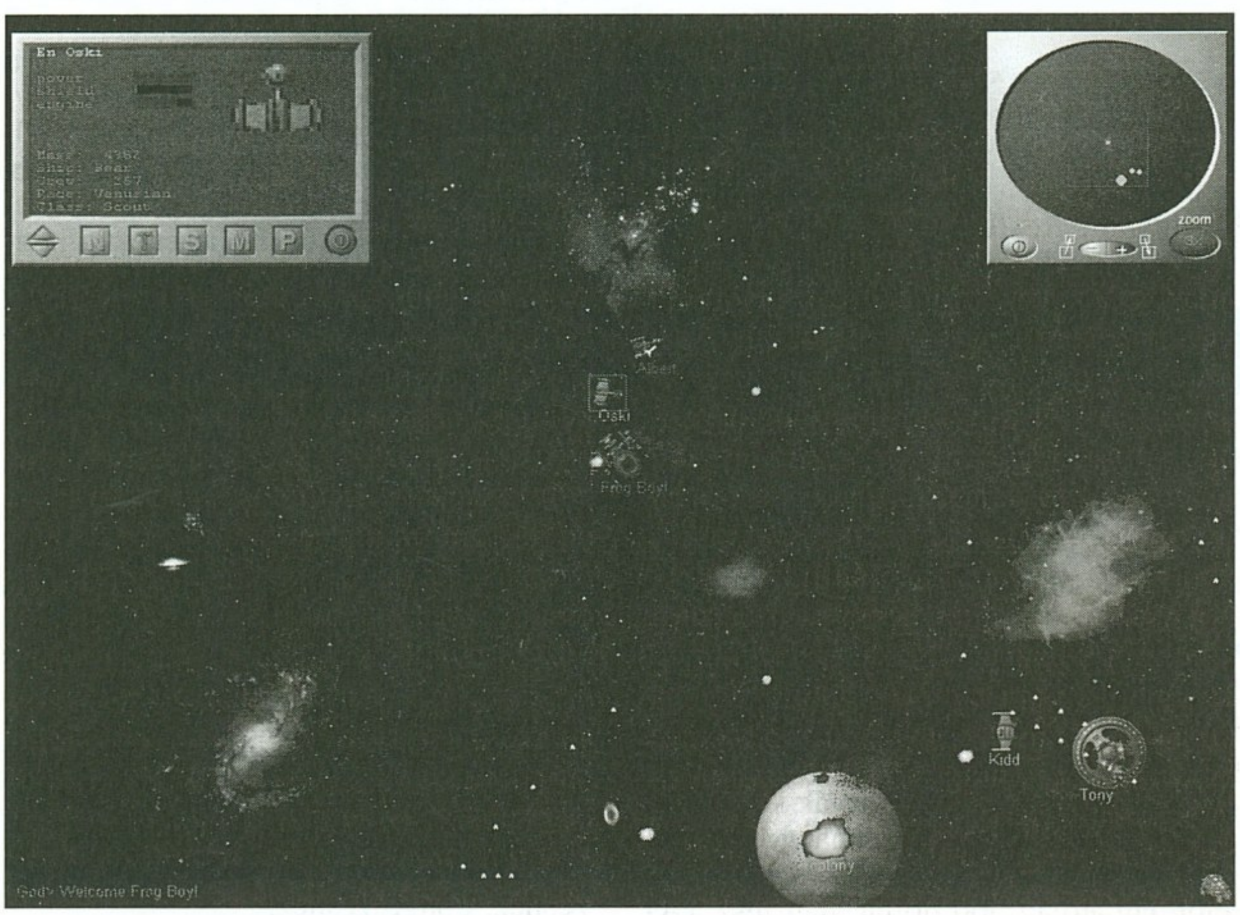
Ismét jöttek a kisebb-nagyobb javítások. Az IBM frissítette a Netware kliens 2.12-es verzióját (ftp://service.boulder.ibm.com/ps/products/netware/Client_OS2_V212/), az 1.1.7-es Java motort (ftp://ftp.hursley.ibm.com/pub/java/fixes/os2/11/117/), és kiadta az első javítócsomagot a VisualAge C++ v4-hez (http://www.software.ibm.com/ad/visualage_c++/service/csd.html) is. Ez utóbbi igen nagy (100 MB), ezért csak a gyors internetes kapcsolattal rendelkezőknek érdemes belekezdeni a letöltésbe. Lényegesen kisebb helyet foglal el az IBM Antivirushoz kiadott frissítés (av30di), amely a Symantec oldaláról (http://www.symantec.com/avcenter/ibm/index.html) szerezhető be. A frissítés telepítése után az Antivirus programcsomag képes lesz az utóbbi időben oly sok kellemetlenséget okozó Melissa vírus felismerésére és irtására is. A nem IBM-es termékekhez megjelent javítások közül az Apache webkiszolgáló 1.3.6-os (http://www.apache.org/dist/binaries/os2/), valamint a parancssor-helyettesítő 4OS2 3.02-es változatát lehet kiemelni (http://www.jpsoft.com/upgrade.htm).



PDF készítés közben

Játékok minden mennyiségben

A szórakozásra vágyók számára jó volt az elmúlt időszak, mert ismét sok új játék, illetve frissítés jelent meg. Teljesen új az OS/2-es játékok körében a LodeRunner program, amely az ftp://cad.ngasu.sci-nnov.ru/pub/OS2/lr.rar címen található. Érdekes módon a dokumentációt oroszul írták, ám a játék menüje (szerencsére) angol. A Stardock által fejlesztett Stellar Frontier játékhoz több frissítés is megjelent néhány napon belül. A legutolsó, a 0.94.7-es verziószámot viselő változat a http://www.stardock.com/products/sf/sf.html oldalon található. Ugyancsak frissítették a már szinte klasszikusnak számító Mastermind játékot (http://www.geocities.com/SiliconValley/Port/4718/), ez jelenleg az 1.41-es kiadásnál tart. Elsősorban a játékfejlesztők számára lehet érdekes, hogy nyilvánossá tették a GameCode multimédia-könyvtár forráskódját. A GameCode eddig csak Win32 és Mac alól volt elérhető, ám a forráskód ismeretében valószínűleg átteszik OS/2-re is.



A Stellar Frontier megújult változata

REXX Tips & Tricks 2.90

Megjelent a REXX tippek és trükkök 2.90-es változatának kiadása. Az OS/2-es gépeken olvasható INF fájlformátumú információs csomag letölthető a http://hobbes.nmsu.edu/pub/os2/dev/rexx/rxtt29.zip címről. A kezdő és haladó REXX programozóknak egyaránt ajánlható adatbázis rengeteg információt tartalmaz magáról a REXX nyelvről, a REXX-hez használható kiegészítő programokról, valamint elképesztő mennyiségű REXX/OREXX példaprogramot is tartalmaz.

SciTech Display Doctor for OS/2

A GRADD driverekkel együttműködő Display Doctor programnak megjelent a harmadik béta-kiadása. A SciTech Software (http://www.scitechsoft.com) által fejlesztett alkalmazás újabb reinkarnációja már sokkal több chipsetet támogat, mint elődjei, így jóval több felhasználó számára jelenthet megoldást a videovezérlőkkel kapcsolatos problémákra. A Display Doctor, amely a hibaelhárításon kívül segít a maximumot kihozni videokártyánkból és monitorunkból, az ftp://ftp.scitechsoft.com/sdd/beta/os2/ könyvtárban található.

AICQ 0.16

Pavel Nikiforov jóvoltából megjelent a szöveges felülettel rendelkező OS/2-es ICQ kliens újabb kiadása. A szokásos hibajavításokon kívül új funkciókkal (SwithBack, PipePreffix, PipePad) is bővült az AICQ repertoárja. Az alkalmazás letölthető az ftp://ftp.intsys.ru/pub/aicq könyvtárból, vagy a Russian/2 kiszolgáló oldalain (http://www.os2.ru/fa/net-cl.shtml) is megtalálható.

Kádár Zsolt

PC, processzor, memória

A Toshiba 64 Mbit-es DRAM termelésének jelentős csökkentését tervezi ez év végig, mert átalakítja a DRAM-kibocsátás szerkezetét. A változtatás a 128 MB-os SDRAM és más nagy teljesítményű memóriák, a Rambus DRAM és a Double-Data-Rate (DDR) SDRAM irányába hat. Decemberre 64 Mbit-es RAM-ból a termelés csak 1 millió darab lesz. Júniustól a Toshiba vezető memóriagyára, a Yokkaichi Works a 0,2 mikronos termékekre állítja át teljes termelését, és a harmadik negyedév végére többi üzeme is ugyanezt teszi.

➢➢

Japánban már megkezdték a Rise mP6 266 (200 MHz-es) processzor árusítását, alig több mint 50 dolláros áron. Ezzel együtt megjelentek a PC 133-as SDRAM-ok, bár hivatalos specifikációt még nem közöltek.

➢➢

A Prosignia notebook-sorozat legújabb tagjaiba a Compaq az AMD K6-2-es processzor 350/380 MHz-es változatait építi be. Jelenleg a hordozható PC-k között ennek az órajele a legnagyobb.

Audio, video, multimédia

Hordozható MP3-as lejátszójának 64 MB-os változatát készítette el a Diamond. A Rio PMP300 (250 dollár) akár 2 óra digitális minőségű zenét vagy 12 órányi beszédalapú hangot képes lejátszani, és kompatibilis az előző változathoz készített 16 és 23 MB-os flash memóriabővítőkkel.

➢➢

A Creative Labs is piacra dobta MP3-as lejátszóját. A Nomad képességei jobbak a Rioénál, beépített FM rádiótunert tartalmaz, és felvenni is tud, tehát egy diktafon funkcióit is ellátja (64 MB-on 4 óra). A 64 MB-os 250 dollárba, a 32 MB-os 170 dollárba kerül.

➢➢

Az S3 és a VIA társul, hogy új chipsetet készítsenek, tovább mérsékelve a PC-k árát és lecsökkentve a beszerelendő chipek számát. A Savage4 és az Apollo Pro chipsetet egyesítő új áramkör kódneve Savage NB, további információk az év második felében várhatók, amikor a két cég formálisan is útjára bocsátja a terméket. (Igaz, az integrált megoldásokat sokan nem kedvelik a bővíthetőség csökkenése miatt, de az átlagos felhasználók már így is sokkal nagyobb teljesítményt kapnak, mint amennyit ki tudnak használni.)

➢➢

Az ATI bejelentette Rage 128 grafikus chipjének továbbfejlesztett változatát, hogy lépést tartson a megjelenő Voodoo 3 és TNT2 konkurrenciával. A Rage 128 Pro processzor 30%-kal gyorsabb órajellel működik, a setup engine teljesítményét megduplázták, ezzel akár 50%-kal nagyobb sebességet érnek el. Ezenkívül támogatja az Anisotroping szűrést (a 3D képsíkkal szöget bezáró felületek minőségét javító eljárás), a textúratömörítést, valamint a DDWG DVI (Digital Visual Interface) minden specifikációjának megfelelően képes kezelni a lapos monitorokat. Az AGP 4x kompatibilis chip mintái már áprilisban elkészültek, a sorozatgyártás júliusban indulhat be.

➢➢

A VideoLogic hosszú szünet után ismét PC-kártyát hoz piacra. A PowerVR Series2 technológiát már egy éve kidolgozták, de először csak a Sega tavaly debütált Dreamcast konzoljában és játéktermi gépeiben kapott helyet. A Neon 250 2D/3D megoldás (szemben a régi PowerVR chipekkel) már a normál 2D műveleteket is tudja, így nem kell mellé külön kártya, 16 MB RAM-mal, 125 MHz-es processzorral van felszerelve, a 32 bites 3D módokat kezeli, AGP 2x kompatibilis. A kártya érdekessége (akárcsak az előző PCX modelleknél) a 32 bites, lebegőpontos Z-puffer (sokszögek távolsághelyes megjelenítéséhez), a számítás teljesítménycsökkenés és memóriafelhasználás nélkül halad. Bár a textúra kitöltési teljesítménye alacsonyabb, mint a Voodoo 3-é vagy a TNT2-é, a speciális architektúrának köszönhetően csak a ténylegesen megjelenő képpontok kerülnek kirajzolásra (a többi kártya azokat is kirajzolja, amelyek takarásban vannak, csak a Z-puffer adatai alapján a később feldolgozott pontok felülírják, komplex 3D modellek esetén rengeteg pluszmunkával). A júniusban várható kártya ára 200 dollár alatt lesz.

➢➢

Az Evans & Sutherland alacsony árú, nagyteljesítményű kártyát jelentett be Lightning 1200 néven. A Pentium III-ra felkészített DynamicGeometry szoftverrel ellátott RealImage alapú kártya 699 dolláros árával a középszintű mechanikai CAD, és az alapszintű digitális tartalomkészítő programok profi felhasználóit célozza meg. A SolidWorks és a Pro/Engineer tesztben 400%-kal, a 3D Studio Max programban pedig 1000%-kal volt jobb, mint egy Riva TNT (Pentium III rendszerben, 1280*1024-es felbontásnál.)

➢➢

Digitális TV-chipet gyárt Janus néven a TeraLogic. A Janus az első egychipes, nagyfelbontású DTVPC dekóder, amely képes az analóg kábeladások és DVD filmek konvertálására, a HD videofelbontásra. A PC-s TV-tunerkártyák között vezető helyen szereplő Hauppauge már bejelentette terveit a chippel, és kártyájuk előreláthatólag 300 dollár alatt lesz. (Az S3 és az Nvidia is együttműködik a TeraLogic-kal a reális árú DTVPC megoldás kidolgozásán.)

➢➢

A 80-as években hazánkban is ismert Paradise visszatér a videokártya-piacra. A három kártya, amellyel Amerikában megjelentek: VCS390 Hawaii 128 AGP (Savage 3D, 8 MB, 99 dollár), Vci740 Bahamas AGP (Intel 740, 8 MB, 50 dollár) és VCS362 Figi Plus AGP (S3 Trio 3D 2X, 45 dollár). Kérdés, hogy tudnak-e pluszt is adni a márkanévhez.

➢➢

Az Nvidia következő, de remélhetőleg még az idén megjelenő chipje támogatja majd a 3D átalakítások és fényhatások számítását, azaz a geometriai számítások nagy részétől felszabadítja a központi processzort, így többszörösére növeli a megjeleníthető sokszögek számát. Ez a módszer nem ismeretlen a CAD-felhasználók számára, a többezer dolláros kártyákon már eddig is volt geometriai processzor. Az Open GL támogatja a geometriai feldolgozást, a Direct 3D viszont csak a következő, 7.0-ás verziótól lesz képes kezelni.

➢➢

A Pioneer 6x-os DVD-meghajtója nemcsak EIDE, hanem Ultra SCSI interfésszel is forgalomba kerül. A 32x-es CD-ROM-nak is megfelelő egység 512 KB puffert tartalmaz, kompatibilis az írható DVD, CD és újraírható CD-formátumokkal, ára 150 (EIDE), illetve 225 (SCSI) dollár.

Érdekességek

Második PC 149 dollárért? Majdnem. A Vega technologies Buddy PC kártyája és szoftvere segítségével a Windows 95/98/NT rendszert használók kettéoszthatják gépüket, csupán egy plusz monitor, egér és billentyűzet kell hozzá. Az egyszerű irodai és otthoni (nem játék) felhasználók számára a cég szerint fel sem tűnik a sebességcsökkenés, mert a jelenlegi processzorok közel sincsenek kihasználva. Az ajánlott gépfelszereltség: Pentium 100 MHz és 32 MB RAM.

➢➢

A Sega Dreamcast konzolja hivatalosan 1999. szeptember 9-én kerül az amerikai boltokba. A 128 bites gép a Sony PlayStationnél 15-ször, a Nintendo 64-nél 10-szer, a jelenlegi Pentium II PC-k grafikus teljesítményénél négyszer gyorsabb, ára azonban csak 199 dollár lesz. Hordozható Visual Memory Unitja nemcsak független játékra alkalmas, hanem játékot cserélhetnek otthon, a játékteremben vagy párban, két VMU-t használva. A konzolok tekintetében a legnagyobb újdonság mégis a beépített 56 K modem teljes hálózati funkcióval, internetes játékhoz, levelezéshez, böngészéshez.

Bánó György


ÖN IS SZERETNE
tisztán látni?
MAGIC-HORSE
FORRADALMI TECHNOLÓGIA
MIRAKEL
Csodatörlő
- mikroszálas
- környezetbarát
- tisztítószer nélkül tisztít
- allergiásoknak nélkülözhetetlen
- élettartama minimum 100 mosás
ARECO
ARECO INFORMATIKAI KFT.
1065 Budapest, Podmaniczky u. 9. Tel.: 302-0158, fax: 331-0340
ARECONET: www.areconet.hu és www.irodaszer.com
Lurdy-Ház: 1097 Budapest, Könyves Kálmán krt. 12-14. Tel.: 456-1256, fax: 456-1255
Rózsakert Bevásárlóközpont: 1026 Budapest, Gábor Áron u. 74-78.
Tel.: 391-5915, fax: 391-5916


PROFON
1138 Budapest, Cserhalom út 4.
Telefon: 350-6227, 350-6235
Telefon/Fax: 350-5093
Számítástechnikai rendszerek komplett hálózatának tervezése és kivitelezése
ADATHÁLÓZATOK, ERŐSÁRAMÚ HÁLÓZATOK, HÍRKÖZLŐ HÁLÓZATOK
RIT HÁLÓZATI ELEMEK
ÖSSZEKÖTŐ KÁBELEK
RACKSZEKRÉNYEK, RACKSZERELVÉNYEK
HÁLÓZATFELÜGYELŐ RENDSZER
ERŐSÁRAMÚ ELOSZTÓSZEKRÉNYEK
TÚLFESZÜLTSÉG-LEVEZETŐK HÁLÓZATOKHOZ, GÉPEKHEZ
SZÁMÍTÓGÉPEK
Rövid kivitelezési határidő, hároméves garancia!
MASTER'S DESIGN

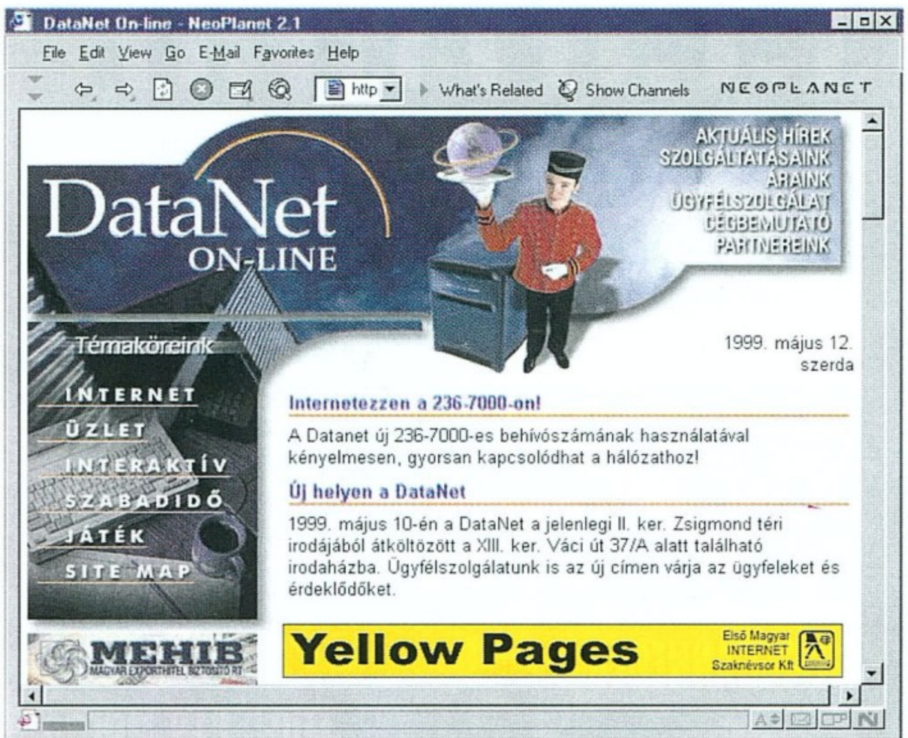

CORG®
COMPUTER
GRAPH ZEPPELIN
DIGITÁLIS VIDEÓ ÉS AUDIÓ
CENTRUM
DIGITÁLIS VIDEÓ
• M-JPEG, MPEG-I és MPEG-II rendszerek széles választékban • Non-linear- és hibrid editorok • 2D és 3D szoftverek, feliratozók • Digitális kamerák, filmszkennerek
DIGITÁLIS AUDIÓ
• Hangkártyák, harddiszkrekorderek, szinkronrendszerek • Effektkártyák, zajcsökkentők • Editáló és szekvenszer szoftverek, hangminták
HÁZI ZENEI STÚDIÓK
MIDI BILLENTYŰZETEK
DIGITÁLIS FÉNYKÉPEZŐGÉPEK
FILMSCANNEREK
MPEG I ÉS MPEG II ENKÓDEREK
DVD VIDEO
CORG Computer Kft. 1111 Bp., Bartók B. u. 46. T.: 466-6675, 386-9810 F.: 319-4899


Ready®
COMPUTERS
READY COMPKER Kft.
Számítástechnikai szaküzlet,
bemutatóterem és szervíz
1054 Budapest, Vadász u. 36.
Tel.: 331-05-18 Fax: 311-86-71
Komplett konfigurációk, alkatrészek, tartozékok, kiegészítők
széles választékával állunk vásárlóink rendelkezésére
Igényei szerint összeállított számítógépek 1+2 év,
MAG, Daewoo, Belinea, Proview monitorok 3 év
garanciával.
300 féle termék, napi akciók:
Abit, Amptron, Aristo, Asus, Procomp, Gigabyte,
Soyo alaplapok. Teljes winchesterválaszték,
HP, Canon nyomtatók, CD-rom, CD-író, VGA kártyák,
hangkártyák, SCSI csatolók, internet szolgáltatások.
Hétfő - Péntek: 9³⁰-18
Szombat: 9-13
Aktuális árlista:
2-333-666/1310#
Internet:
www.ready.hu
ready@alarmix.net


DIGITÁLIS FÉNYKÉPEZÉS
CÍMKE/FELIRAT NYOMTATÁS
Digitális fényképezőgépek, PC kamera, film/dia scanner, lapscanner, nyomtató, memóriakártya, kártyaolvasó, PCMCIA adapter, akkumulátortöltő, akkumulátor, fotópapírok tintasugaras nyomtatókhoz.
Nem zsákbamacska!
Nézze meg, próbálja ki!
6/9/12/18/24/36 mm széles laminált, vízálló, színes, öntapadó szalagok.
Önállóan és számítógéppel használható típusok.
M11 ipari betűnyomó alumínium és rozsdamentes acél szalagokkal.
DYMO M-11
Nikon
brother
P-touch
LettraSoft
Budapest, 1149
Egressy út 5.
T./f.: 221-6779
221-6772
Győr, 9024
Mónus I. u. 19.
T./f.: 96/414-411
Fax: 517-501
DIT
DIGITALTECHNIKA
Kodak
digital science
LS500
ipari nyomtató
29/57/100 mm
széles,
színes, öntapadó
szalagok
kültérre/beltérre.

Virtuális bevásárlóközpont

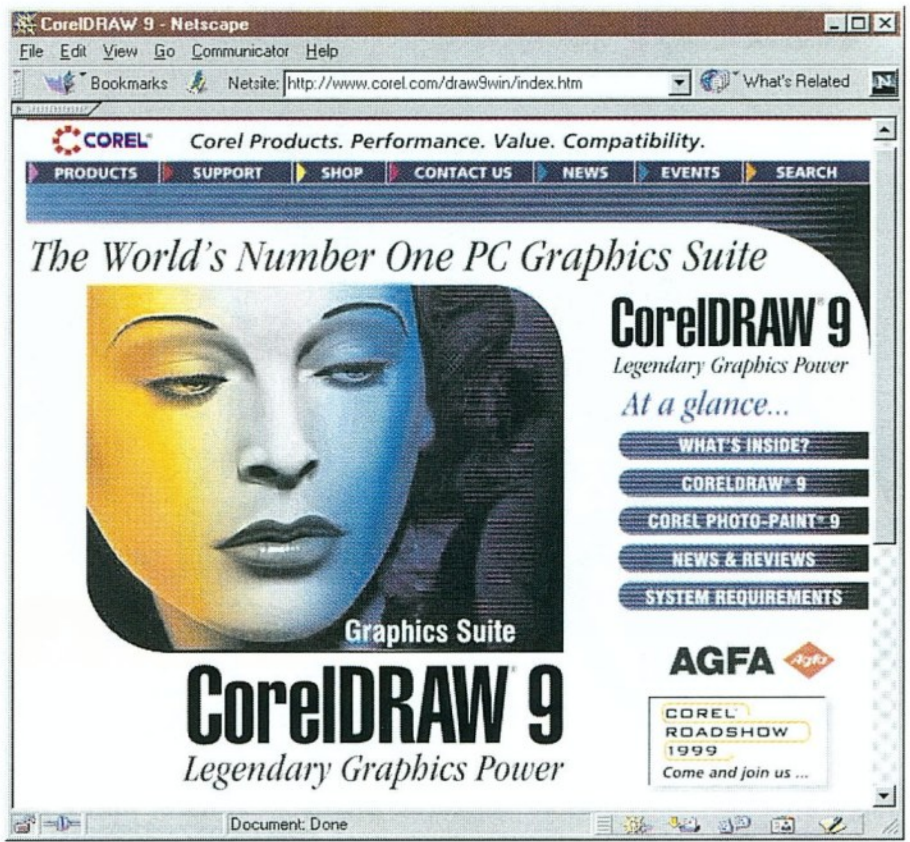
Az IBM és az Inter-Európa Bank közel egy éve jelentette be az első SET (Secure Electronic Transfer) szabványon alapuló internetes kereskedelmi szolgáltatás megindítását, illetve a hasonló külföldi rendszerek hazai viszonyokra történő adaptálását. 1999 áprilisában megnyitották a Passage Virtuális Bevásárlóközpontot, amelynek kínálata a DataNet közreműködése révén válik hozzáférhetővé az internetes kereskedelem számára. A „Passzázsban” termékeiket elhelyező cégeknek a DataNet megfelelő szoftvereket biztosít, melyekkel akár saját áruházi osztályok is kialakíthatók. A rendszer kliensprogramjától elektronikus pénztárcát „kapunk”, és ebből költekezhetünk, ha a virtuális áruház polcain valami megtetszik. Az elektronikus kereskedelem biztonságát az adja, hogy az áru kiválasztását követően a vásárló a számla kiegyenlítése idejére „átkerül” a banki rendszerre, tehát személyes banki adatait sem az Internet szolgáltató, sem az eladó nem ismerheti meg. A vásárlási tranzakció azonban az eladó és vevő között zajló folyamat marad, a kettőjük közötti



kölcsönös bizalom alapján. Az Internet szolgáltató legfeljebb közvetett felügyeletet gyakorolhat, például kizárhatja azokat, akikkel szemben túl sok megalapozott vásárlói reklamáció merül fel.

Alkalmazott „Psionológia”

A Magyar Autóklub „sárga angyalok” segélyszolgálatának híradástechnikai infrastruktúrája jelentősen korszerűsödött. A Pannon GSM, a Psion Rendszerház Kft és a K&H Bank közreműködésével a mobil egységek és a diszpécserközpont között ezentúl rövid szöveges üzenetek (SMS) „utaznak”. A Pannon GSM által biztosított szolgáltatás segít kideríteni a segélykérők hollétét, majd a javítást végző visszajelzést ad a munka befejezéséről. A maximum 160 karakterből álló üzenet megfejtését a segélykocsikba telepített Psion POS (eladáshelyi) terminál végzi. Ez a mobil számítógép a szerelők adminisztratív munkáját is segíti, mert a háttérben folyamatos nyilvántartást vezet. Az esetleg felmerülő többletköltségeket a bajba jutott autós akár a bankkártyájáról is kiegyenlítheti a helyszínen.

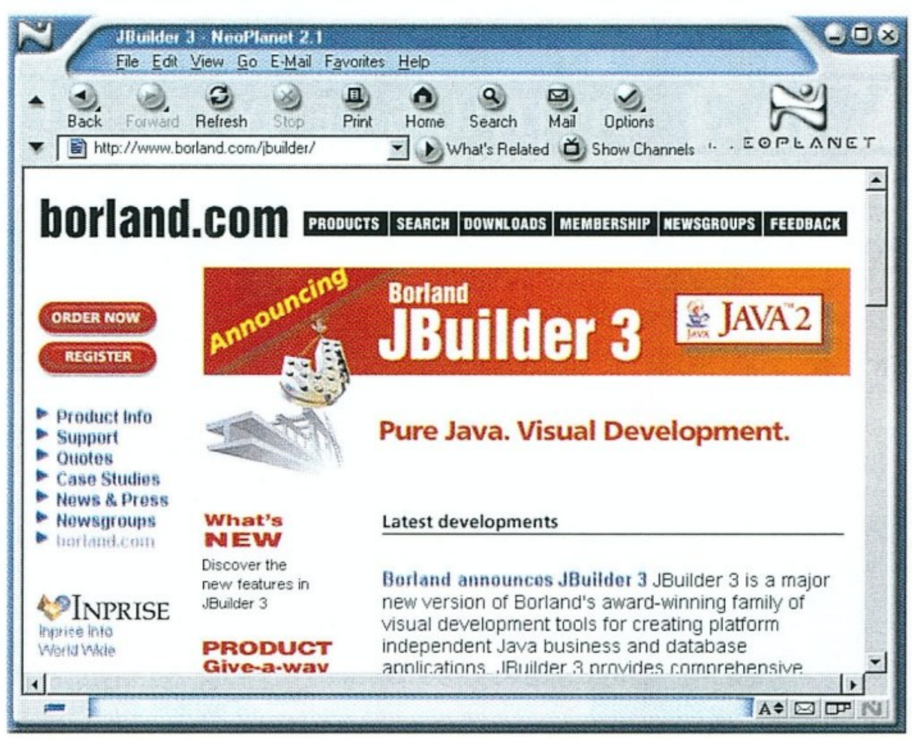


Úton a CorelDraw 9

Lezárultak a Corel népszerű grafikai programjának új verzióján végzett bétatesztek. A végleges változatot az Info 99-en, a Codra standján meg lehetett tekinteni működés közben. A CorelDraw 9.0 piaci megjelenése a Corel honlapja szerint (www.corel.com) hamarosan várható, lehet, hogy mire lapunk megjelenik, ez már meg is történik. Az új változat minimális környezeti igényei: Win95/98/NT4.0 rendszer, 32 MB RAM (de 64 MB a javasolt), Pentium 133, CD-ROM meghajtó, legalább 100 MB szabad hely a merevlemezen.

Elkészült a JBuilder 3

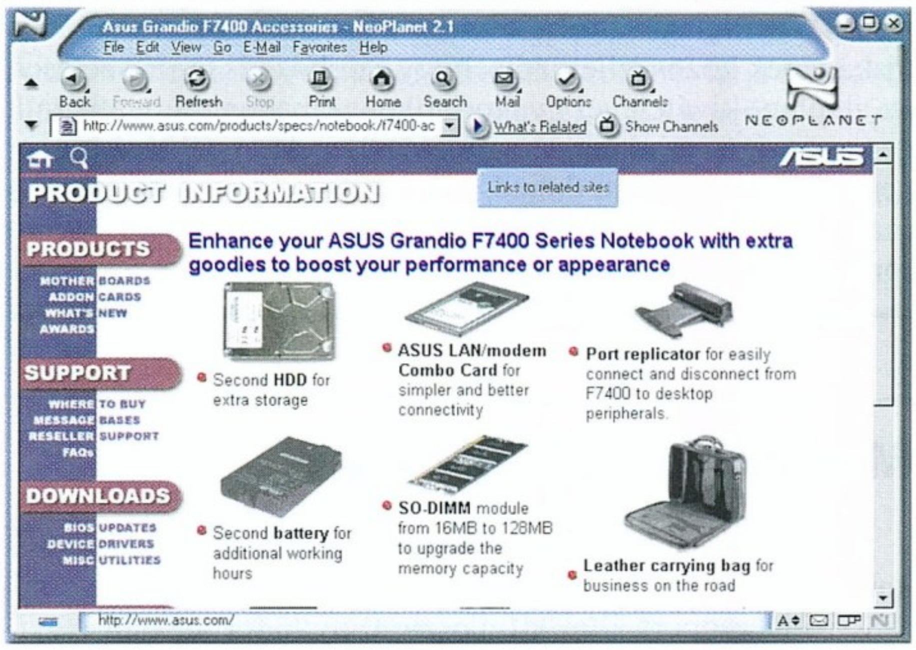
Az Inprise Corporation Borland részlege 1999 április közepén jelentette be a vizuális Java fejlesztési eszköz, a JBuilder új verzióját. Mivel a Borland JBuilder alapvetően a Sun JDK verziókra alapozott termék, segítségével mód van platformfüggetlen Java fejlesztésekre, bár a fejlesztőgéptől ez több erőforrást igényel. A JBuilder 3 nyitott környezete



támogatja a JDK 1.1.x-et, a JFC/Swing komponenseket, a JavaBeans és Enterprise JavaBeans elemeket, a Corba, RMI és JDBC alapú kapcsolatokat, valamint ezek mellett a legtöbb vállalati adatbázisszervert. Megvan a támogatása a Java 2 platformon is, mely szintén kedvez a sokféle gépen való fejlesztésnek és futtatásnak. A piaci tervek szerint az új JBuilder először a „hagyományos" 32 bites Windowsra jelenik meg, és azt a Solarisra készült változat követi, még ebben az évben. Az Inprise tervezi a teljes rendszer megjelentetését Linuxra is. A termék az előzőekben kialakult „méretezést" követve Enterprise, Professional és Standard változatban kapható.

Digitális iroda

Az Info '99-en érdemes volt körülnézni olyan szempontból is, hogy a digitalizálódással foglalkozó májusi számunkban megrajzolt tendenciák hogyan jelennek meg a nagy nemzetközi cégek magyarországi kínálatában. Mintha a hazai fizetőképes kereslet kicsit lomhán mozdulna e trendek irányába, és ez érződik a „seregszemle" másik oldalán is. Nem sok csorgott át ide az őszi Comdexen elindult digitális offenzívából. De azért volt mit megnézni, és volt min elgondolkodni. Legvonzóbban talán a Minolta tálalta azt a koncepciót, hogy a nyomtató, a szkenner és a fax legyen egyetlen kompakt rendszerben (lásd az alábbi képkompozíciót), és a digitális fotózásban és képmanipulálásban is komoly fejlesztési eredményekkel álltak elő. A hardver oldaláról a digitális forradalom magabiztosan nyomul előre, a szoftverek (és az emberek) ezzel nem mindig tudnak lépést tartani.



Új Asus táskagépek

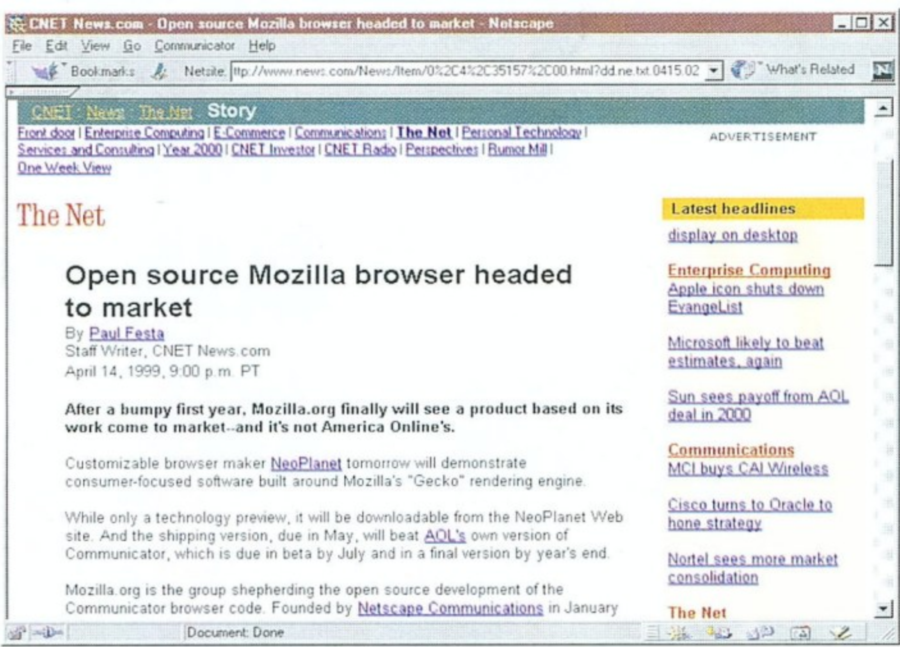
Az Asus név jól ismert azok körében, akik személyesen foglalkoztak PC-jük belső összetevőivel, például az alaplappal. Több lábon állni azonban biztonságosabb, ezért az Asus kiterjesztette a fejlesztést a táskagépekbe (notebook) építhető



alaplapokra is. Az asztali gépekkel ellentétben azonban az Asus a notebook alaplapokat csak saját gépeibe fogja beépíteni. Magyarországon ezeket a táskagépeket a Sowah Kft forgalmazza, és 1999 áprilisában meg is kezdte a legújabb típusok bemutatását, elsőként az F7400-as gépét. Az Asus táskagépek közös jellemzője, hogy a doboz (és természetesen az alaplap) kivételével gyakorlatilag szabadon válogathatjuk össze az igényeinknek megfelelő elemeket, például a processzort, a memóriát, de még a kijelzőt is. Az F7400-asokba már DVD-lejátszót is beépíthetünk (http://www.asus.com /products/specs/notebook/f7400-acc.asp).

Mozilla és NeoPlanet

A Netscape felvásárlásakor sokan kifejezték aggodalmukat, hogy az America Online esetleg „nem fogja kedvelni" a Mozilla csoport által kifejlesztett, nyílt forráskódú Navigatort. Az AOL azonban biztosította az Internet közösségét a nagy múltú böngésző „Gecko" munkanevű, szabad kódú új verziójának támogatásáról, amit azzal is alátámasztott, hogy a Navigator 5.0 motorját éppen a Mozilla fejlesztéséből kívánta beépíteni. Ugyanakkor beigazolódott az is, hogy a forráskód felszabadítása a két nagytól független fejlesztőket intenzívebb munkára serkentheti. Májusi CD-nken már közzétettük a NeoPlanet 2.0-s változatát. Az újabb verziók a Gecko motorját is tudják majd használni. A jelenlegi csak az


CNET News.com - Open source Mozilla browser headed to market - Netscape

File Edit View Go Communicator Help

Bookmarks Netsite: ttp://www.news.com/News/Item/0%2C4%2C35157%2C00.html?dd.ne.txt.0415.02 What's Related

CNET : News : The Net Story

Front door | Enterprise Computing | E-Commerce | Communications | The Net | Personal Technology | Services and Consulting | Year 2000 | CNET Investor | CNET Radio | Perspectives | Rumor Mill | One Week View

ADVERTISEMENT

The Net

Open source Mozilla browser headed to market

By Paul Festa
Staff Writer, CNET News.com
April 14, 1999, 9:00 p.m. PT

After a bumpy first year, Mozilla.org finally will see a product based on its work come to market--and it's not America Online's.

Customizable browser maker NeoPlanet tomorrow will demonstrate consumer-focused software built around Mozilla's "Gecko" rendering engine.

While only a technology preview, it will be downloadable from the NeoPlanet Web site. And the shipping version, due in May, will beat AOL's own version of Communicator, which is due in beta by July and in a final version by year's end.

Mozilla.org is the group shepherding the open source development of the Communicator browser code. Founded by Netscape Communications in January

Latest headlines
display on desktop

Enterprise Computing
Apple icon shuts down EvangeList

Microsoft likely to beat estimates, again

Sun sees payoff from AOL deal in 2000

Communications
MCI buys CAI Wireless

Cisco turns to Oracle to hone strategy

Nortel sees more market consolidation

The Net

Document: Done

Internet Explorer böngészőmotorját látja el esztétikusabb és kényelmesebb kezelői felülettel. (Az eddigi verziókból a CNet híre szerint közel 1 milliót töltöttek le, ami akkor is tekintélyes szám, ha tudjuk, hogy nem minden példány honosodik meg véglegesen a gépeken.)

A Silicon Graphics 3 arca

A Silicon Graphics a nagy teljesítményű 3D grafikus munkaállomásokkal szerzett hírnevet. A „high tech", a teljesítmény, a minőség fogalma kötődött hozzá, és — valljuk be — a „hétköznapi halandók" számára egy kicsit az elérhetetlensége is. Időközben azonban lezajlottak olyan szerkezeti változások, melyeket a kívülállók nem nagyon akartak észrevenni, ezért a 17 éves cég most határozott lépést tett

Ez egy valódi Silicon Graphics munkaállomás



arculatának árnyaltabbá tételére. A vállalat új neve az eddig tőzsdei és internetes rövidítésként használt SGI. Termékei három fő kategóriába tartoznak, és ezt a márkanevek különbözősége is jelzi. A Silicon Graphics elnevezést használják továbbra is a vizuális munkaállomásokra (az árbevétel felét azért még mindig ez a termékcsoport teszi ki). A vektor-szuperszámítógépek családja visszakapja a Cray nevet, míg a szerverek SGI márkanéven kerülnek forgalomba (www.sgi.hu).

Automatikus zárás Linuxon

A HRP Hungary Kft-től az Info '99-en kapott információk szerint az MGE UPS Systems is csatlakozott a Linux-felhasználókat támogató gyártók csoportjához. A Personal Solution-Pac (PSP) for Linux biztosítja a Linux szervereket használók számára, hogy áramkimaradás esetén is adatvesztés nélkül megtörténjen a folyamatban lévő számítástechnikai műveletek lezárása. Programozható az áthidalási idő, egy parancsfájl segítségével pedig megadhatók a lezáráskor szükséges lépések. A PSP kompatibilis az RS232-es soros csatlakozójú MGE UPS rendszerrel, amely a Pulsar EL sorozat kivételével a teljes Pulsar családot jelenti, a nagyobb UPS-eknél viszont szükség lehet az RS232/U-Talk kártyára.

Hibrid szoftvervédelem

Bár a szoftverek illetéktelen használatának megakadályozására alkalmazott szoftveres és hardveres védelmi megoldások a „közszükségleti" termékeknél gyakorlatilag teljesen eltűntek, néhány nagy értékű professzionális programnál ma is alkalmazzák a másolásvédelmet vagy a hardverkulcsot. Az Aladdin Knowledge Systems magyarországi partnere, az Onyx Kft legutóbb egy hibrid megoldást mutatott be, amely ötvözi a szoftveres és hardveres védelem lehetőségeit. Az Info '99-en is bemutatott HASP rendszer tartalmazza a gép

kimenetére csatlakoztatható hardverkulcsot, benne az azonosításhoz szükséges mikrokontrollert vagy ASIC (Application Specific Integrated Circuit) chipet. Futásidőben az alkalmazás ezek segítségével tudja lekérdezni a státuszt. Szoftveroldalon ennek függvényében vagy aktivizálódik az alkalmazás, vagy legális megerősítés hiányában alternatív működés, például demó üzemmód lép érvénybe. Az így védett alkalmazásokhoz indulócsomagként rendelkezésre áll a HASP Developer's Kit a hardverelemen kívül tartalmazza a fejlesztéshez szükséges dokumentációt és a programot. (Kérdés, hogy ki jelenik meg, és mi történik, ha megdörzsöljük Aladdin lámpását.)

Linux és a BSA

Az Info '99 kiállításon bukkant fel a plakát, amely első ránézésre a BSA és a Linux-pártiak közös akciójának tűnt, de amint a kibocsátó MIME (Magyar Internetezők Magánegyesülete) honlapjáról (http://www.euroastra.com) kiderült, nem volt köztük ilyen egyeztetés. Tekintsünk el most a formális ügyektől, és próbáljuk meg a plakátot tartalmilag megközelíteni. Abból, hogy a BSA hogyan viszonyul hozzá, választ kaphatunk egy nagyon kényes kérdésre. Arra ugyanis, hogy kit és mit képvisel a BSA. Amennyiben a plakátot saját céljaival egybevágónak tekinti, akkor a BSA valóban a szerzői jog tiszteletben tartásának prioritása mellett teszi le a voksot. A plakát eszmei mondanivalója ugyanis az, hogy tartsd tiszteletben a szerzői jogokat, és ha nincs pénzed arra, hogy megfizesd az Üzleti Szoftver Szövetség (BSA = Business Software Alliance) tagjai által gyártott és forgalmazott szoftvereket, akkor ne kockáztasd a kilátásba helyezett büntetést, hanem használj inkább ingyenes operációs rendszert (amilyen például a Linux), és ezen a platformon futtatható alkalmazásokat (amilyen például a StarOffice). Amennyiben a BSA saját érdekeinek sérelmét látja az alábbi plakátban, akkor egyértelmű az üzenet: a BSA nem a szerzői jogot védi, nem is az alkotómunkát végző programozók érdekét, hanem a vállalkozók üzleti érdekét. Ez utóbbit természetesen szintén kell védeni, csak nem valami gusztusos a szerzői jog paragrafusait erre használni (angolul: abuse). De minderről részletesebben is olvashatnak majd szeptemberi számunkban.

Excelt olvasó Java

A táblázatkezelők kimeneti állományát többnyire csak különböző konverziókkal tudjuk a Weben közreadható formába hozni. Az Excel esetében magához a konverzióhoz általában kell a Windows, ha pedig a konverzió OLE alapú, akkor gyakran maga az Excel is. A Tidestone Technologies által kifejlesztett Formula One for Java az Excel táblák kezelését Java környezetben is megvalósítja: írja és olvassa, valamint létre is hozza a Microsoft Excellel kompatibilis táblákat. A webmestereknek Java appletként, illetve fejlesztőknek JavaBeanként rendelkezésére álló Java termék használatához a Sun JDK 1.2 is szükséges. (A szerverhez és a felhasználói oldal használatához a licencet külön kell megvásárolni.)

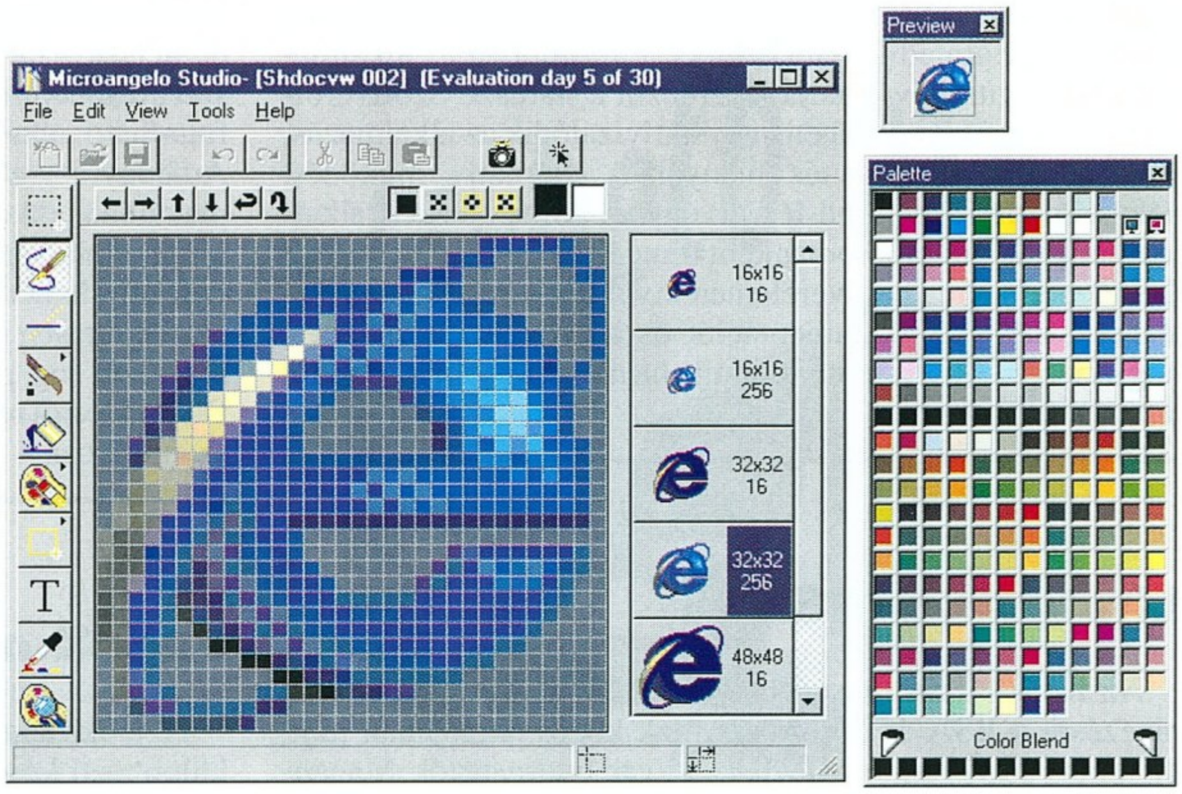


A Business Software Alliance (BSA) börtönbüntetéssel fenyegetőző plakátja

Tux, a szabadon terjeszthető Linux operációs rendszert reprezentáló pingvin

File Commander II

Név: File Commander II v2.01
Platform: DOS, OS/2, Windows 95/98/NT
Szerző: Brian Harvard
Honlap: http://www.silk.apana.org.au/fc.html
Licenc: shareware
Korlátozás: nincs
Regisztrálási díj: 25 dollár



Ismét egy Norton Commander klón. Sem nem új, sem nem forradalmi, nekem mégis a mai napig a kedvencem, és ennek több oka van. Először is kicsi és gyors. A program kibontva 460 Kbájt (a windowsos 970 K), de a tömörítő programokat (unzip, unarj, unrar) mellé téve is simán felfér egy lemezre. Nincs szükség telepítésre, bárhonnan, bármikor könnyedén futtatható. Szöveges módú, ezért gyors, ám közben kiválóan együttműködik a grafikus munkafelülettel. Támogatja az egér használatát, képes gyűjtőként külön megnyitni az aktuális könyvtárat, egy mozdulattal helyezhetünk árnyékot az aktuális objektumról a munkaasztalra. Rendelkezik saját vágólappal, de használhatja a rendszer vágólapját is. Beépítetten támogatja a legelterjedtebb tömörített formátumokban (ZIP, ARJ, RAR, LHA, ACE) készült fájlok kezelését, sőt ha a megfelelő tömörítő program elérhető, akkor egyszerűen lehet vele ki- és becsomagolni. Támogatja a hosszú fájlnevek használatát, beépített szövegszerkesztővel rendelkezik, képes több lépésben viszaállítani a korábbi állapotot (undo), kódlapok között tud váltani. Rendkívül rugalmasan konfigurálható, megjegyzi a könyvtárakat, ahova azután egy mozdulattal visszatérhetünk (vagyis nem kell folyton végigmászni az egész könyvtárstruktúrán). És ez az egyetlen (általam ismert) commander klón, amelyik a folyamatállapot kijelzésekor a teljes folyamatállapotot jelzi, nem pedig egyenként a fájlokat, amivel nem sokra megy az ember. Olyan shareware, amelyben nincs sem funkcióbeli, sem időbeli korlát, csak néha feszólít a regisztrálásra.

Ja, és (majdnem elfelejtettem) létezik OS/2-es és windowsos változata is, természetesen mind a kettő 32 bites. Egyetlen komoly negatívuma, hogy nem képes hálózat kezelésére (ahogy a FAR). Meg persze nem tud CD-t vagy MP3-at lejátszani, nincs benne MPEG dekóder, sőt tetris sem. De sebaj.



Microangelo 98

Név: Microangelo 98
Fájlnév: mike98.exe
Platform: Windows 95/98/NT
Fejlesztő: Impact Software
Honlap: http://www.impactsoft.com/
Licenc: Shareware
Korlátozás: 30 napos időkorlát
Regisztrálási díj: 59,95 dollár

Az operációs rendszerek fejlődése során bebizonyosodott, hogy a használhatóság szempontjából nagyon fontos elem a grafikus munkafelület. Ez az, amelyen keresztül a felhasználó kapcsolatot tart az alkalmazásokkal, a gép erőforrásaival és a rendszer elemeivel. Természetesen minden rendszer felülete különböző, vannak olyan elemek is, amelyek mindenütt megtalálhatók. Ilyenek például az ikonok, amelyek a rendszerek különböző elemeit, alkalmazásait vagy adatállományait reprezentálják. Rendkívül sokfélék. Akadnak köztük csinosak és kevésbé igényesen kivitelezettek, vannak amelyekkel gyakran találkozunk, másokkal csak rendkívül ritkán. Van, aki számára lényegtelen, hogy milyenek a rendszer ikonjai, de olyanok is akadnak, akik szeretik minden részletét tekintve testre szabni a munkafelületet. Ez utóbbiak számára készült a Microangelo.

Az ikonszerkesztés igazából nem bonyolult feladat, a kis méretekből adódóan egyeseknek az sem gond, ha pi-

xelenként kell megrajzolni. A képpontonkénti szerkesztésen túl egy korszerű ikonszerkesztőnek azonban egyéb problémákkal is szembe kell néznie. A klasszikus ikon 32x32 képpont nagyságú, de a legtöbbnek létezik 16x16-os kicsinyített változata, illetve néhányhoz van nagyobb, 40x40-es változat. Ezenkívül egy ikonnak mindegyik méretben lehet 16, 256, 65 ezer és 16 millió színű (true color) változata. (Hogy mi értelme 16 millió szín használatának, amikor egy ikon jellemzően 4096 képpontból áll, és semmiképpen sem tartalmazhat ennél több különböző színű képpontot, azt fedje a jótékony homály...) A különböző változatok kezelése szerencsére ebben a programban jól megoldott. Nagyon jó az egyes módok (például 16x16 vagy 32x32 méretű) közötti konverziós algoritmus. Egyszerűek, de jól használhatók a rajzeszközök (ceruza, ellipszisrajzolás stb.), jól átlátható és használható a felület.

A Microangelo ugyanakkor nem csupán ikonszerkesztő. Tartalmaz egy eszközt, amellyel a rendszer ikonjait le lehet cserélni. Ez a Windows Explorerhez hasonló ikonkezelő, amely csak ikonokat és ikonokat tartalmazó állományokat jelenít meg. Segítségével a rendszer bármely eldugott zugában könnyen rá lehet akadni az ikonokra, sőt lehetővé teszi a DLL-ekben található ikonok szerkesztését is. Az egyetlen általam tapasztalt negatívum, hogy az erőforrásokban található ikonokat nem lehet külön elmenteni. Tartozik még a Microangelohoz egy mozgó ikonszerkesztő, illetve kurzorszerkesztő kiegészítés, valamint egy ikonkönyvtár-nyilvántartó segédeszköz is.

Mindent összevetve a Microangelo komplett és kiforrott megoldás azok számára, akik örömüket lelik az ikonok lecserélgetésében és átszerkesztésében.

WindowBlinds



Név: WindowBlinds
Fájlnév: wblind80.zip
Platform: Windows 95/98/NT
Fejlesztő: Stardock Systems
Honlap: http://www.stardock.com/
Licenc: shareware
Regisztrálási díj: 19,95 dollár

Sokan állítják, hogy a Windows sikerének egyik legfontosabb eleme a tetszetős megjelenés volt. Lehet, hogy így volt, lehet, hogy nem, saját szubjektív véleményem szerint a Windows még így is elég csúnyácska. Aki Linuxot használ, az már megszokhatta, hogy különböző X Window felületek variálgatásával egyéni ízlésének megfelelően átalakíthatja a rendszer kinézetét, és az OS/2 Warp is jónéhány segédeszközt kínál a megjelenés felforgatásához. Windows alatt azonban valamilyen okból viszonylag kevés a felület testreszabását célzó alkalmazás. Pedig, hogy lenne min változtatni, arra jó példa a Stardock Systems WindowBlinds nevű programja.

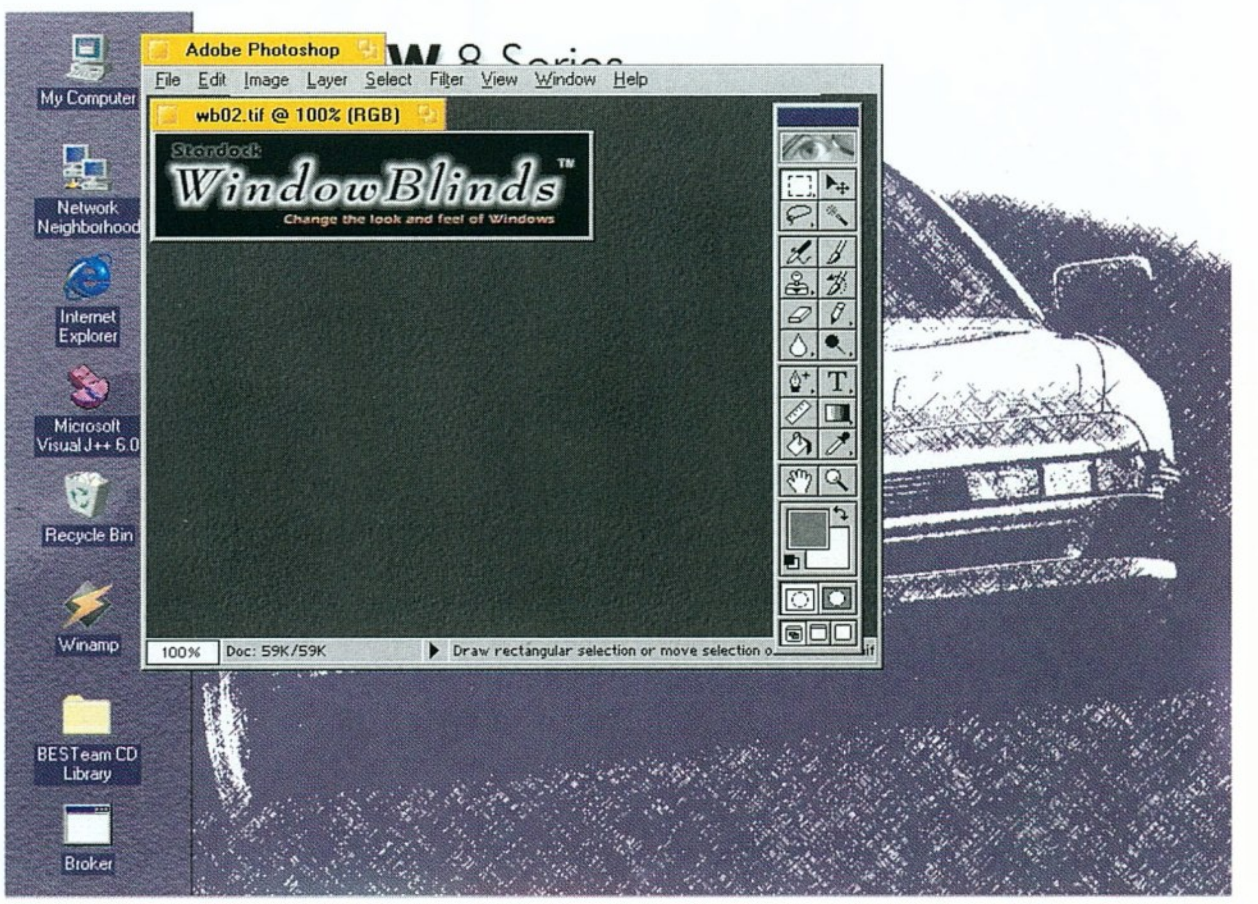
Egy ablakorientált grafikus operációs rendszerben, amilyen a Windows, az ablakok általában szabványos elemekből — gombokból, keretekből, háttérekből stb. — épülnek fel. Ezen elemek annyira jellemzőek, hogy az operációs rendszer pusztán ezek alapján is könnyedén felismerhető. Feltéve, hogy nem fut rajtuk WindowBlinds. Ez a program ugyanis éppen ezen grafikai elemek lecseréléséről gondoskodik.

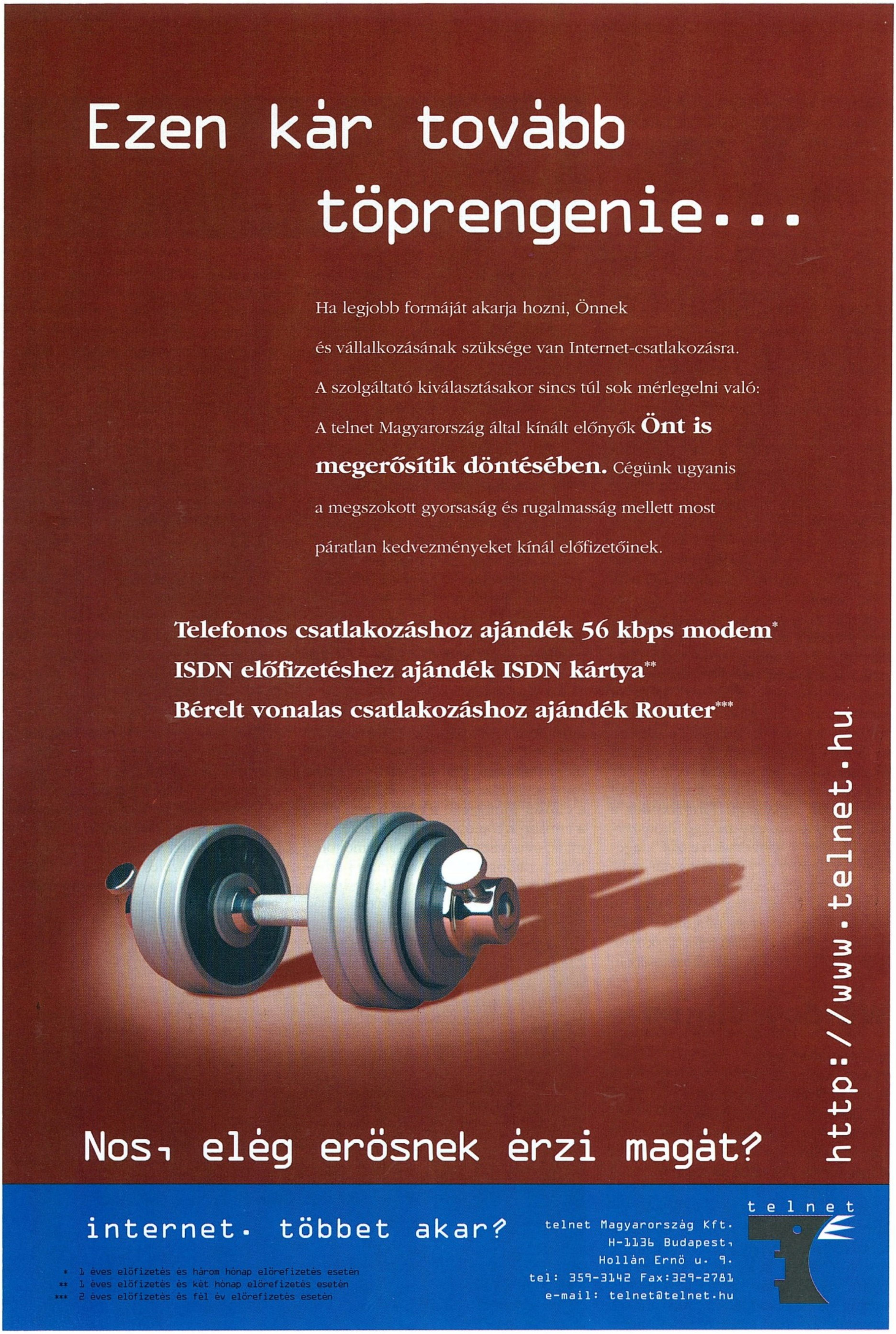
Az már szinte természetes, hogy a programba eredetileg beépített megjelenéseken kívül bárki készíthet egyéni ízlésének megfelelő felületet (skint). A megjeleníthető grafikai elemek majd mindegyike lecserélhető a segítségével: a címsor mintázata, a címsor alakja(!), a címsori nyomógombok kinézete, formája, elhelyezkedése(!), a nyomógombok, a rádiógombok formája, a párbeszédablakok stb. Gyakorlatilag teljesen átszabható a rendszer. Szépek a programban eleve meglévő minták, de sok szép minta tölthető le az Internetről is.

És most a negatívumokról. A program elég bizonytalan. Néhány ablaktípust elfelejt átalakítani (például a hagyományos szöveges ablakokat), és egyesekkel összeakad. Bár a programban megadott kivétellista segítségével megelőzhetők a lefagyások (a kivételnek deklarált alkalmazás kinézetét nem bántja a program), így viszont egyre kevesebb az olyan ablak, amelyen egyáltalán megjelennek a változtatások. És végül — bár ez kissé szubjektív — nekem nem tetszik, ha a gépemen folyton ott ül egy program, amely semmi mást nem csinál, mint arra vár, hogy minden ablak kinézetét megváltoztassa. Sokkal szimpatikusabb volna, ha a rendszer erőforrásai cserélődnének fel (az alapértelmezett, beépített képek és függvények), és a megjelenítéshez nem kellene külön program, bár lehet, hogy ehhez olyan mélyen kellene beletúrni a rendszerbe, amire esetleg csak a Microsoft vállalkozhatna. És persze az is igaz, hogy ez így sokkal rugalmasabb.

Összegezve: az ötlet remek, a kinézet nagyon szép, az instabilitás remélhetőleg megszűnik, amint a program továbblép az előzetes (0.80-as) verzióról.

Nagy Tamás





INFORMÁCIÓKÉRÉS: 25 ▲

Madge: IP alapú videomegoldás

A Madge Networks piacra dobta „Video in a Box" elnevezésű videokommunikációs LAN megoldását, amellyel a felhasználók IP alapú videoalkalmazásokat tudnak telepíteni lokális hálózataikban. Az ún. induló kit egyszerű megoldást nyújt ahhoz, hogy a felhasználók LAN/WAN videokommunikációt kezdeményezzenek. A Video in a Box tartalmazza a Madge LAN Video Gateway eszközét Ethernet vagy TokenRing hálózatra, továbbá négy H.323 (IP) és egy H.320 (ISDN) desktop videokonferencia (codec) egységet. A LAN Video Gateway szabvány alapú együttműködést biztosít IP bázisú videorendszerek és meglévő ISDN alapú videokonferencia rendszerek között. Rétegelhető architektúrája következtében a Video Gateway a videohálózat igény szerinti növekedését úgy támogatja, hogy megengedi extra egységeknek a hálózathoz való kapcsolását. A Video in a Box LAN készlet ára Európában 15-19 ezer dollár között van.

Ericsson: IP telefónia

Az Ericsson IP telefónia megoldása, az IPT rendszer, amely telefon–telefon, fax–fax és PC–telefon kapcsolati szolgáltatásokat támogat a TCP/IP hálózatokon. Az IPT kihasználja a számítógép megnövelt teljesítményének és továbbfejlesztett kompressziós technikájának előnyeit, s ennek révén valós idejű alkalmazások használatát teszi lehetővé a nagy kapacitású IP hálózatokon. Az Ericsson IPT rendszer főbb elemei a következők. A telefonhálózat és a TCP/IP hálózat közötti hidat jelentő Voice Gateway; a PSTN és IP alapú hálózatok közötti szignalizációs információk átvitelére szolgáló SS7 Gateway; Interactive Voice Response szolgáltatás; az IPT forgalomirányítója és adatmenedzsere (Netkeeper és Sitekeeper); számlázási adatbázis és account server; az ún. 0&M rendszer; a webböngészőben megvalósított grafikus menedzsment; PC kliens szoftver a TCP/IP hálózatból kimenő hívásokra. Az IPT rendszerben számos helyet kötnek össze TCP/IP hálózattal. Minden egyes hely a kapcsolt telefonhálózattal is össze van kötve. A Netkeeper és a Sitekeeper fizikai TCP/IP interfészei Ethernet 10Base-T vagy 100Base-T típusúak lehetnek. Az operációs rendszer Windows NT 4.0 és ENEA OSE Real Time Operating System.

Automatizált intranet megoldás

Az Areco Systems Kft Java-kompatibilis technológiát használó dokumentummegjelenítő alkalmazást kínál intranetekhez. A Net-It Central v.3.5 változatú szoftver lehetővé teszi a dokumentumok automatikus összegyűjtését és publikálását. A programcsomag használatakor nincs szükség arra, hogy a dokumentumokat előzetesen HTML formátumúra konvertáljuk, így jelentős idő és költség takarítható meg. A szoftver szerver alapú, így a könyvtárszerkezet központilag történt kialakítása után a felhasználónak csak be kell másolnia az eredeti formátumú (Word, Excel stb.) állományokat a megfelelő könyvtárba, és a szoftver automatikusan, akár előre megadott időpontban publikálja a könyvtár tartalmát. A kliens oldalon a dokumentumok standard böngészőben plug-in nélkül tekinthetők meg. A rendszerkövetelmények a következők: Win95 vagy NT szerver, minimálisan 32 Mbájt RAM, 10 Mbájt szabad terület; a forrásdokumentumok tárolhatók bármely (Win95, NT, Unix, Novell, Mac) fájlszerver platformon.

Internetes tartalomszűrés

A nem kívánt Internet tartalom szűrését lehet elérni a Secure Computing cég SmartFilter programcsomagjával, amely egyben az Internet hozzáférés költségét is csökkentheti. A szoftver a SmartFilter Control List URL adatbázist használja. A programcsomag használatával a következő funkciók valósíthatók meg: a haszontalan tartalom kiszűrése (a rendszeradminisztrátorok 27 kategóriából tetszőleges típusú URL-ek szűrését állíthatják be); prioritások beállítása (az üzleti tevékenységhez alapvetően szükséges oldalak letöltése előnyben részesíthető a listán szereplő oldalakkal szemben); testre szabott korlátozások adott munkaállomásokon; a SmartFilter Control List automatikus letöltése FTP-vel; testre szabható hibaüzenetek (a tiltott oldalak lekérésekor megjelenő hibaüzenet tetszőleges formát ölthet); üzleti órák megadása; bizonyos UseNet hírcsoportok kiszűrése; hyperlog készítése (az internetes hozzáférés trendjei könnyen kiértékelhetők); testre szabható jelentések készítése és teljes körű naplózás. A SmartFilter számos tűzfallal képes együttműködni, de tud a tűzfal mögött elhelyezkedő helyettesítő kiszolgálóként (Proxy Server) is üzemelni. A felhasználók webböngészője minden WWW és FTP kérést a SmartFilter Proxyhoz irányítanak, az analizálja a kéréseket, és az aktuális vállalati biztonsági politika szerint engedélyezi vagy tiltja az adott oldal elérését.

Novell termékek az IBM és az Oracle világában

Megjelent a NetWare for SAA 4, az IBM környezetekre fejlesztett LAN-host kapcsolati megoldás nyílt bétaváltozata. A leendő közös Novell-IBM termék révén NetWare hálózatok, vállalati intranetek és Internet megoldások integrálhatók IBM nagyszámítógépekkel és AS/400 rendszerekkel. A NetWare for SAA 4 teljes mértékben tartalmazza az NDS címtárat és termékként a NetWare 5-tel integráltan fog megjelenni. A szoftver támogatja a TCP/IP és SNA protokollokat. Másik hír szerint a Novell megállapodott az Oracle-lel, hogy az NDS-t az Oracle 8i adatbáziskezelővel integrálják, ami által biztosítható, hogy az Oracle Internet Directory (OID) és az NDS együttműködjön. Az együttműködés eredményeként egyszerűsödhet az Oracle alkalmazások adminisztrációja, továbbá a két címtár integrációjával lehetővé válik, hogy az Oracle alkalmazások az „NDS fán" elhelyezkedő összes hálózati erőforrást (például hálózati kapcsolókat, desktop alkalmazásokat, szervereket stb.) OID interfészeken át elérjenek. További Novell hír: a cég forgalomba hozta a NetWare 5 Network File System (NFS) Services programcsomagját, amely Unix alapú munkaállomások és Novell alapú számítógépek között létesít kétirányú kapcsolatot. Ezzel a termékkel a vegyes környezet az NDS-t használó egyetlen pontról menedzselhető. A NetWare 5 NFS Services a WebNFS protokoll 2.0-ás változatát támogatja. Ennek eredményeként a WebNFS képességű böngészővel rendelkező felhasználók Internet vagy vállalati intranet révén érhetik el a NetWare 5 szerveren lévő állományokat. A NetWare NFS Services a Novell Storage Services (NSS) köteteket is támogatja, ezáltal a felhasználóknak kisebb tárigényt és az NSS kötetek felépítéséhez nagyobb sebességet biztosít.

Lucent: vezeték nélküli új LAN

Vezeték nélküli helyi hálózatának új, nagy sebességű változatát jelentette be a Lucent Technologies. Az április közepén megtartott budapesti sajtótájékoztatón bemutatott WaveLAN/IEEE Turbo elnevezésű termék Ethernet sebességű hozzáférést nyújt az Internethez és az üzleti alkalmazásokhoz. A turbo kivitel az IEEE 802.11 szabvánnyal kompatibilis jelenlegi vezeték nélküli LAN rendszerek áteresztőképességének háromszorosát éri el. Az új rendszer speciális automatikus sebességválasztási funkciót is tartalmaz, amely az egyes környezetekhez önállóan kiválasztja az optimális üzemmódot. Akik jelenleg a WaveLAN-t használják, azok számára a Lucent a WavePoint II Access Point PC ikerkártyás architektúrát ajánlja a turbó technológiára való áttéréshez. Az új WaveLAN/IEEE kártya turbó és IEEE üzemmódban egyaránt képes működni.

Kovács Attila

Novell fejlesztői készletek

A nyitottságban az erő

Az OSA (Open Solutions Architecture) a fejlesztések keretrendszere, a fejlesztőket segítő stratégia. A Novell ennek kialakításával a gyártói fejlesztéseken túlmutató keretet kívánt biztosítani a külső fejlesztés számára, melynek igénybevétele iránt az egyre komplexebbé váló hálózatokon fokozódik az igény, nemegyszer harmadik féltől igénybe vett fejlesztésként, mert a speciális, egyedi felhasználói igények nem mindig építhetők be az alaprendszerbe. De nem is lehet reális cél minden potenciális igény kielégítését egy konkrét termékbe, például egy hálózatos operációs rendszerbe zsúfolni, mert ez esetleg indokolatlanul bonyolultabbá teszi a rendszert, megnehezíti az áttekintést, megnöveli a lehetséges hibaforrások számát.

Számos fejlesztőműhelyben meghonosodott gyakorlat, hogy az általános feladatokat ellátó programhoz publikálják azokat a csatlakozási felületeket, esetleg programfejlesztési segédleteket, melyek lehetővé teszik a külső fejlesztések zökkenőmentes csatlakoztatását az alaprendszerhez.

A Novell hálózatos programjaihoz felhasználhatóan számos információt tett már közzé korábban is, az OSA szellemében a különböző termékek megjelenésével többé-kevésbé párhuzamosan. Az említett segédletek összességét tartalmazza a Novell Developer Kit, mely ennek következtében nem egy konkrét fejlesztési segédprogram, hanem egyfajta forrás, belépési kapu az OSA alapú fejlesztésekhez. A rendszeresen frissített információkat az Interneten a http://developer.novell.com/ndk/ címen érhetjük el, melyen 1998 augusztusa óta publikálják az időszakonként CD-n is megjelenő csomagokat, információkat.

A **Novell Developer Kit (NDK)** a korábban elmondottaknak megfelelően összefogja azokat a fejlesztői segédleteket is, melyek már korábban elérhetők voltak a különböző Novell termékekhez. Az OSA elveinek megfelelő egységes csatlakozási modell biztosítja a menedzselhetőséget az egyes alkalmazások elkészítésekor, a többi hálózati elemhez való kapcsolódáskor, és az üzleti vagy nagyobb hálózatokon történő futtatásakor.

A Novell termékeihez fejlesztő programozóknak összeállított csomagokat (kiteket) több nagyobb csoportba sorolhatjuk. A szoftverfejlesztési csomagok **(Software Developer Kits — SDK)** biztosítják azokat a segédleteket (utility), könyvtárakat, dokumentációs anyagokat, mintakódokat és nyílt API-kat (Application Programming Interface), amelyek segítik a programok könnyű elérését (az operációs rendszertől az Internetig), illetve a teljes rendszer menedzsmentjét. Támogatást adnak a programnyelveken végzett fejlesztésekhez is, a C/C++ nyelvtől a szkriptnyelvekig.

A támogatottak közül kiemelten fontosak azok az esetek, amikor az Internet technológiáját alkalmazzák a hálózatban. Például egy IP-alapú intranet esetében a Java osztálykönyvtárak és komponensek (JavaBeans) és a JavaScript támogatása kiemelt szerepű.

A hardverfejlesztési csomagok **(Hardware Developer Kits — HDK)** szolgáltatják azokat a mintakódokat, dokumentációkat, melyeket a különböző meghajtók fejlesztésekor használhatunk fel. Így a szoftveresen heterogén hálózatok mellett a különböző hardverelemek hálózatos kapcsolatait is biztosíthatjuk, például erőforrásként felvehetjük az NDS fába. Azonban az ide tartozó elemek a kompatibilitási tesztekben is szerepet kapnak.

A fejlesztési segédleteket tartalmazó csomagok **(Developer Tools Kits — DTK)** szerepe hasonló az SDK-éhoz, pontosabban azok jobb kihasználásához adnak segítséget. Ennek elemei tartalmazzák a Novelltől vagy harmadik féltől származó, de a Novell által tesztelt eszközöket, amelyekkel az SDK-ba tartozó csomagokat hatékonyabban használhatjuk fel. Így a különböző fejlesztési platformokat támogató eszközök megtalálhatók itt is.

A kompatibilitási tesztekhez szintén találunk segítséget és eszközöket az NDK keretein belül **(Compatibility Test Kits — CTK)**. Ezekben vannak alkalmazások, tárolóegységek, LAN és WAN, valamint helyi hálózatos rendszerszolgáltatások és printerszolgáltatások teszteléséhez szükséges anyagok. A tesztek következetes elvégzése mindenképpen ajánlott ahhoz, hogy egy új, a NetWare hálózathoz integrálni kívánt fejlesztés eredményeként megbízható alkalmazást kapjunk. De különösen szükségesek ezek a tesztek akkor, ha mi magunk szeretnénk a fejlesztésekkel megjelenni a piacon.

Összességében a Novell Developer Kit univerzális interface a programozók és a programokat tesztelők, minősítők, végső soron az azokat felhasználók számára, bármely olyan platformon, melyhez a NetWare alapú rendszerek kapcsolódási pontokkal rendelkeznek. Ezek pedig felölelik a NetWare operációs rendszerrel telepített szervereken kívül a Windows NT és Unix szervereket, illetve a különböző rendszerekkel felszerelt munkaállomásokat. A heterogén rendszerek kapcsolatának megvalósításában a Novell is kínál különböző megoldásokat, és az NDK is támogat többféle programozási megoldást, de ezek között az igazi közös nevező legtöbbször a NetWare 5-ben már gyárilag beépített Java alapú megoldások alkalmazása.

Simay Endre István



Hazai Linux konferencia

1999. május 31-én Budapesten, a Petőfi Csarnokban rendezik meg az első országos Linux konferenciát és kiállítást. A rendezvény szervezője az ETB Hungary Kft, szakmai támogatója a Linux Felhasználók Magyarországi Egyesülete. Várhatóan több mint egy tucat előadás hangzik el, és nemzetközi cégek is képviseltetik magukat, így jelen lesz előadással az IBM, az Oracle, a Hewlett-Packard és a Sun. (www.linux.hu és linux.inetcomm.com)

Szünetmentes Microsoft

Az American Power Conversion (APC) és a Microsoft közös kampányt folytat, melynek keretében minden olyan magyarországi vállalat, amely 1999. április 1. és június 30. között Microsoft Windows NT szervert vásárol, kap egy APC SmartUPS szünetmentes tápegységet és egy magyar Windows NT szerver Resource Kitet.

PanTel telefon

Elsőként a PanTel kapott engedélyt Magyarországon IP telefonszolgáltatásra. Ennek alapján a PanTel a távbeszélőközpontok béreltvonali hozzáférésén és az Internet hálózaton keresztül nyújthat telefonösszeköttetéses szolgáltatást, a „rendes" telefonhálózat igénybevétele nélkül.

A Btrieve karrierje

A Magic adatbáziskezelő magyarországi disztribútora, az Onyx Szoftverház 1999 márciusa óta képviseli a Pervasive Software céget is, mely még 1982-ben, SoftCraft néven kezdte meg a Btrieve adatbáziskezelő fejlesztését. Azóta a Btrieve kalandos utat járt be. 1987-ben a Novell vette meg, és őt adta a NetWare szerverekhez. Amikor fejlesztését a Novellnél abbahagyták, az eredeti fejlesztőműhely vásárolta vissza, az pedig 1996-ban nevet változtatott. A Magic használóinak azért ismerős a Btrieve, mert annak kliens oldali változatát a dobozban megkapták. A folyamatos fejlesztés, illetve az időközbeni felvásárlások eredményeként készült el egy komplex programcsomag, a Pervasive.SQL. (A terméknév és a fájlnév keverésének lehetősége nyilván sok barátot szerez majd nekik.) Az Onyx egyik legutóbbi rendezvényén Brian McAuliffe, a fejlesztő cég képviseletében mutatta be a közepes méretű adatbázisokkal dolgozó vállalkozásokat megcélzó programot, melynek alapját a Btrieve 7.0 adja. A fejlesztést a Scalable SQL és az ODBC meghajtók új verziója segíti. A programegyüttessel ugyanaz az adattábla tranzakciós és relációs műveletekkel egyaránt kezelhető, ami leegyszerűsíti a maximum 64 GB-os adatbázisfájlok kezelését. Az alkalmazások fejlesztésekor lehetőség van közvetlen beágyazásra a rendszer API-ján keresztül, és ActiveX-alapú elérésre is. Az adatok publikálása internetes formában is történhet. Az Internet felé fordulást mutatja a cég részéről a Tango megvásárlása. A Pervasive Software már be is jelentette a Tango Objects for Macromedia's Dreamweaver 2 csomagot a weblapok szerkesztéséhez.

Az Intel és a divat

Ki hitte volna, hogy az a cég, amely tudatunkban szinte egybeolvadt a számítógép „lelkével", egyszer majd elkezd intenzíven érdeklődni a számítógép „teste", sőt öltözködése iránt. Az Intel 1998. szeptembere óta formatervező (design) cégekkel és a PC-iparból verbuvált partnerekkel együttműködve azon dolgozik, hogy forradalmasítsa a PC-k külső megjelenését. Szó, ami szó, elkelne némi kreativitás ezen a területen, bár aki kénytelen időnként szerelgetni a „gépházban", az nem annyira a divattervezőket, mint inkább az ergonómusokat (és azok rokonságát) szokta emlegetni. Igaz persze, hogy a mostani tervezők sem csupán arra gondolnak, hogy egyéni színeket és formákat vigyenek a formatervezésbe, hanem az egyszerűbb használhatóság, a jobb kezelhetőség is szempontjaik között szerepel. Készségesen elhisszük — ha majd látjuk.

Oracle TechNet

TechNet webhelyet létesített az Oracle. Itt egy ingyenes azonosító megszerzése után Linuxon futó Oracle szerverekről szóló információkhoz lehet hozzájutni. A http://technet.oracle.com/ címről az Oracle 8i vagy az Oracle Web Application Server teljes funkcionalitással bíró linuxos próbaváltozata is letölthető, illetve megrendelhető.

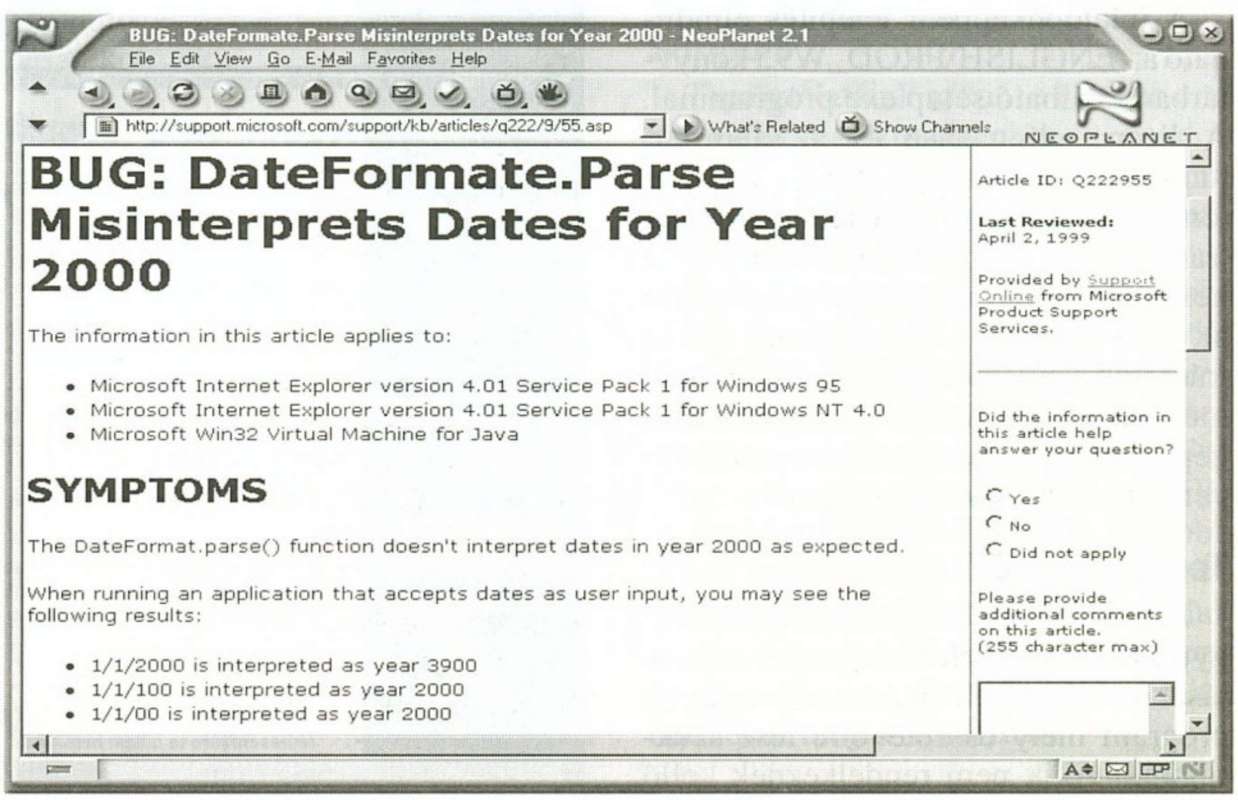
Megint egy érv az átállásra

A Microsoft elismerte, hogy a DateFormat.parse( ) funkció hibájának „köszönhetően" az MS Internet Explorer 4.01 32 bites Windows szervizcsomagja és a Virtual Machine for Java korábbi verziója bizonyos dátumbeírásokat helytelenül kezel, például:

1/1/2000 esetén az évszám 3900

1/1/100 esetén az évszám 2000

A felsorolásban az is szerepel, hogy 1/1/00 esetén az évszám ugyancsak 2000, de ez tulajdonképpen korrekt, mert az időszámításnak (sajnos) nincs nulladik éve, és ha megszokásból az évszám két utolsó évét írjuk be, jelen körülményeink között talán mégis az a legpraktikusabb, ha a program a duplanullát a 2000. évként fogja fel. A hiba javítására a Java motor újratelepítését javasolják az 5.00.3167.1 belső verziójú (3167-es build) MSJAVX86.EXE segítségével. Ez a mostani CD-mellékletünkön megtalálható a Java SDK kíséretében, mivel a Microsoft néhány más csomagtól, például a Windows 98 javítócsomagjától eltérően széles körben szabadon letölthetővé tette.



Kicsit bonyolult, de inkább csak szokatlan

A StarOffice regisztrálása

Lapunk előző három számában a StarOffice 5.0 irodai alkalmazáscsomag ingyenes egyéni használatú (Personal Edition) változata szerepelt CD-mellékletünkön, mégpedig Linux (március), 32 bites Windows (április) és OS/2 (május) platformra. E három hónap alatt a telepítéssel, de főleg a regisztrálással kapcsolatban sokan kértek tőlünk tanácsot telefonon, levélben vagy e-mailben, ezért a felvetett kérdések alapján készítettünk egy összefoglalót, remélve, hogy abból a programra utólag felfigyelők is minden szükséges eligazítást megkapnak.

Az alábbiakban a konkrét telepítési információk a Windows felületen működő változatra vonatkoznak, a regisztrálással kapcsolatosak viszont általában mindegyikre érvényesek.

Kicsomagolás

A StarOffice 5.0 for Windows telepítőcsomagjai tömörítve kerültek fel a CD-re a 32 bites rendszer hosszú fájlnevei miatt. A csomagnak a winchesteren való kicsomagolásához is a hosszú fájlneveket érintetlenül hagyó kibontási technika szükséges, ez megvan az újabb keletű fájlmenedzserekben (például a Windows Commanderben) vagy az új tömörítőkben. (A CD-n a SZERSZAM\FESZER\ZIP könyvtárában megtalálható a winzip70.exe program.) A kicsomagolt telepítőkészletet külön átmeneti könyvtárban célszerű elhelyezni (például C:\TEMPSO), majd telepítés után ezek az állományok letörölhetők.

Adatok

A tulajdonképpeni telepítés elindítható az ENGLISH\PROD_W95 könyvtárban található setup.exe programmal. A licenc elfogadása és az alapvető tudnivalók elolvasása után ki kell tölteni egy űrlapot. Azon feltétlenül beírandó a vezetéknév (last name) és a keresztnév (first name), a címből pedig az utca (street), a város (city) és az irányítószám (zipcode). E-mail címünk megadása is kell, ha a regisztrálást gépünkről a Világhálón keresztül közvetlenül le tudjuk bonyolítani.

Telepítési mód

A telepítéshez szükséges egy 16 jegyű kulcs. A telepítési folyamat elindulásakor meg kell határozni, hogy a program mely összetevőire lesz szükségünk. Akik nem rendelkeznek kellő telepítési rutinnal, válasszák inkább a Standard Installation pontot, ez 138,8 MB lemezterületet igényel, a telepítés idejére átmenetileg pedig 157,4 MB-ot. Akinek kevesebb szabad helye van, az válassza a Minimum Installation menüpontot, 85,4 MB lemezterület kell hozzá, telepítés alatt átmenetileg pedig 87,1 MB. Utána a célterület megadása következik, és a telepítés géptípustól függően néhány perc alatt lezajlik. A telepítésben járatosak választhatják a Custom telepítési módot, így mindent saját igényeikhez igazíthatnak.

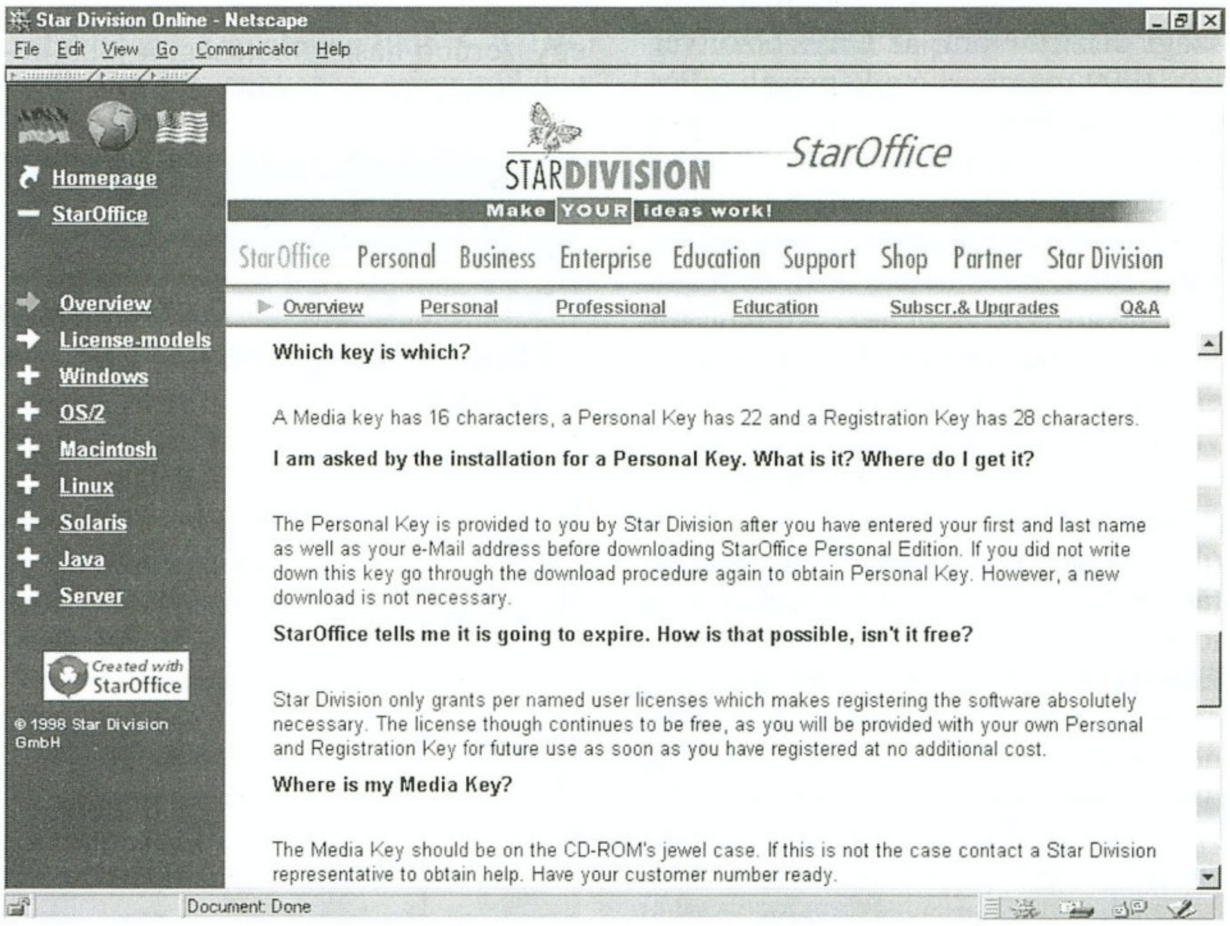
Regisztrálás

A StarOffice regisztrálás nélkül 30 napig használható. A program induláskor felajánlja a regisztrálás elindítását. Ha nemmel válaszolunk, akkor később is bármikor kiválaszthatjuk a Help menüből a Registration pontot. A regisztrálást négyféleképpen lehet elvégezni:

a) Közvetlen Internet kapcsolattal

Ha gépünkön közvetlen Internet elérés van (például modemen keresztül), akkor az Online Registration pontot kell választani. A megjelenő űrlapon lévő információk helyességét ellenőrizni kell, azokat szükség esetén módosítani lehet, de később az alapvető információkon (név, cím stb.) már nem szabad változtatni. Ha előzőleg nem tettük volna meg, el kell végezni az internetes beállításokat (Internet Settings), majd az adatokat el kell küldeni a Star Divisionnak (Submit Data). Ezt követően rövid időn belül e-mail válaszban megkapjuk a 28 jegyű regisztrálási kulcsot (Registration Key) és ügyfélszámunkat (Customer Number). Az e-mailben lévő adatok segítségével a StarOffice szükség esetén (például ha letöröltük) bármikor újratelepíthető és regisztrálható, nem kell a Star Divisionnal a kapcsolatot ismét felvenni.

A pontos kitöltés nagyon fontos, mert aki esetleg nem tudná, az online regisztrálást automatikusan végzi egy erre beprogramozott szoftver, amely „buta" módon egyetlen betűnyi elütést sem tolerál, tehát a pontatlanság megakaszt-



ja a regisztrálási folyamatot. Más szempontból nézve viszont a regisztráló rendszer nagyon is kifinomult, mert a telepítőkulcsot (Media Key) és a kitöltött űrlap adatait felhasználva készíti el a végleges egyedi regisztrációs kulcsot. A CD tasakjának hátoldalán található telepítőkulcs leolvasásakor gyakori tévesztési lehetőség a nulla (0) és a nagy O betű összekeverése. (E két karakter az írógépeken sokáig azonos volt, a számítógépeken azonban kezdettől fogva megkülönböztetjük őket egymástól.) Előfordulhat a tasakok összecserélése is, márpedig ha nem az adott platformnak megfelelő kulcsot írjuk be, azt a rendszer természetesen nem fogadja el, hanem hibaüzenetet küldözget.

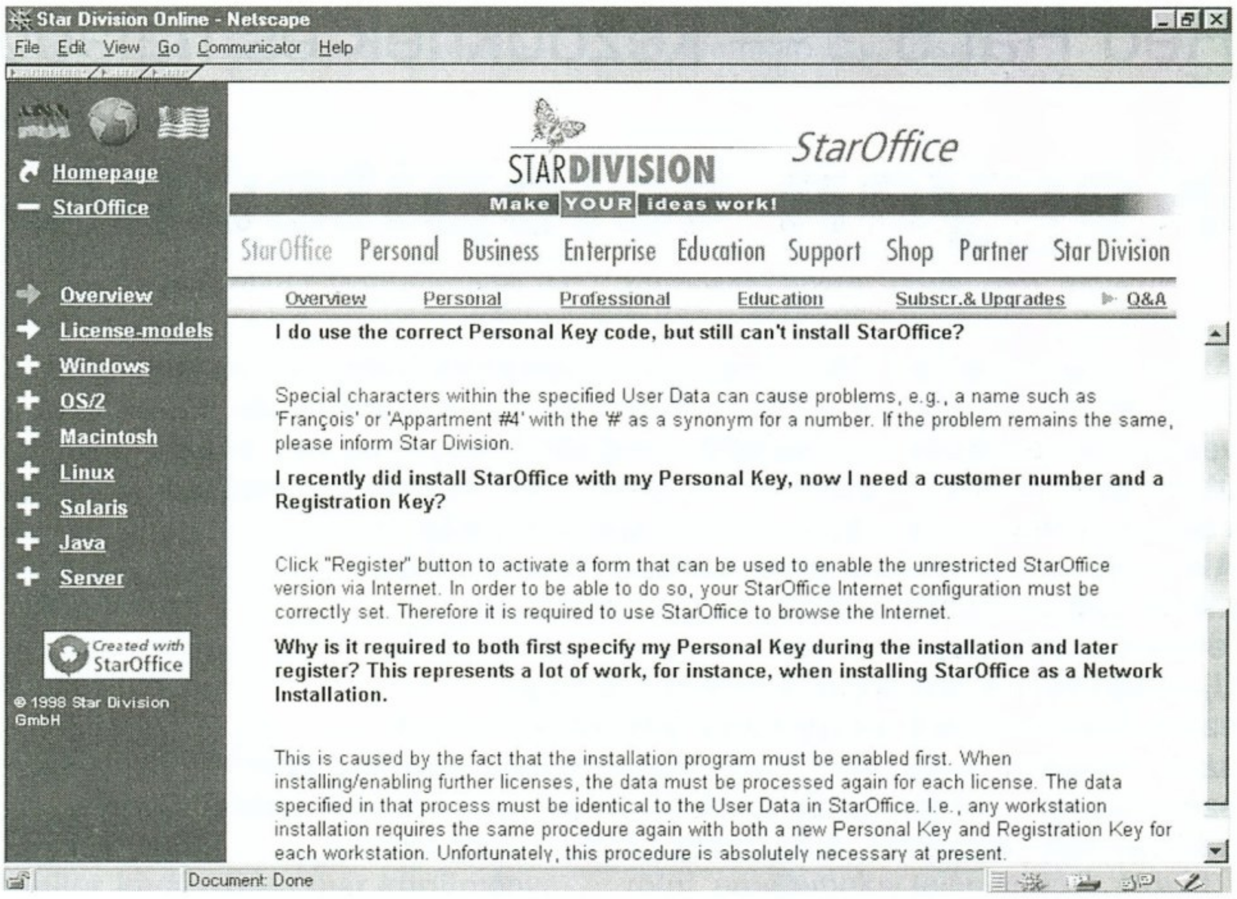
b) Közvetett Internet kapcsolattal

A másik regisztrálási lehetőség az ún. Web Registration, ami tulajdonképpen azt jelenti, hogy az a számítógép, amelyre a StarOffice telepítése történt, nem kapcsolódik a Világhálóhoz (nincs online kapcsolata), másutt azonban van alkalmunk az Interneten keresztüli regisztrálásra. A Web Registration opcióban még módosíthatunk az űrlap adatain. Utána az űrlapot ki kell nyomtatni (Print the Web Data).

Lényeges dolog, hogy nem elegendő a képernyőről lemásolni az adatokat, mert ott nem jelennek meg azok az átmenetileg szükséges kulcsszámok, amelyeket a rendszer a másutt történő regisztráláshoz generál (Check Key, Personal Key). Akinek nincs nyomtatója, az mentse el az űrlapot fájlba (a StarOffice felkínál ehhez egy saját lemezikont is). Ebből a fájlból a szükséges kódszámok szintén kiolvashatók. A regisztrálási cím a http: //www.stardivision.com/registration /english, vagy a német weblap (http: //www.stardivision.de). A 28 jegyű regisztrációs kulcsot általában néhány percen belül megkapjuk.

c) és d) Faxon vagy postán

Akinek egyáltalán nincs lehetősége az Internet használatára, az faxon vagy levélben bonyolíthatja le a regisztrálást (Register per fax, Mail registration). Az eljárás első fele mindkét esetben ugyanaz, az űrlapot ki kell nyomtatni. Utána azonban a kinyomtatott lapot el kell faxolni (00/1/510/5051475) vagy postázni kell a Star Division címére (Star Division Corp. 6515 Dumbarton Circle, Fremont CA 94555, USA). A választ és benne a regisztrációs kulcsot ennek megfelelően faxon vagy levélben kapjuk meg. A regisztrálás és a program további használata is teljesen ingyenes.



A StarOffice használatával kapcsolatban olvasóink számos kérdést intéztek hozzánk, de ezekre majd későbbi számainkban térünk ki, az alábbiakban csak olyasmit érintünk, ami általános jellegű, illetve a szoftver installálásakor, regisztrálásakor, használatba vételekor merült fel.

Korábbi számok

A legtöbb kérdés a korábbi számok beszerzésére vonatkozott. Lapunk nem rendszeres olvasói közül sokan ugyanis csak utólag értesültek arról, hogy teljesen ingyen hozzájuthatnak egy igen jó képességű irodai szoftvercsomaghoz. Az [Új] Alaplap (szerencsére) azon kiadványok közé tartozik, melyek példányait évekre visszamenőleg is keresik — és megveszik. Tudtuk, hogy a StarOffice-t tartalmazó három szám iránt az érdeklődés az átlagosnál jóval nagyobb lesz, ezért azokból többet is tartalékoltunk.

Magyar piac, magyar nyelv

A második leggyakoribb kérdés az volt, hogy mikor lesz készen a StarOffice magyar nyelvű változata. A Star Division eredetileg 1999 első negyedévére tervezete (több kelet-európai nyelvre is) a honosítást, de amikor 1998-ban ingyenessé tette az egyéni használatot, a program iránt keletkezett rendkívül nagy érdeklődés és az új piaci koncepciónak megfelelő szervezet felállítása rengeteg többletmunkát okozott, ezért a határidők sajnos kitolódtak. Ugyanez az oka annak is, hogy a StarOffice professzionális változatának értékesítési csatornáit Magyarországon még mindig nem építették ki.

Tartalékkulcs

Az egyedi installáló kulcs a CD-melléklet tasakjának hátoldalára ragasztott címkén található, de az persze elveszhet, megsérülhet... néhány példányról esetleg le is maradt. Mindez azonban nem akadálya a telepítésnek, mert szerkesztőségünkben van elég tartalékkulcs, azokból nyugodtan kérhetnek másikat, akár telefonon is.

Macintosh és Solaris

A StarOffice 4.0 verziójának volt Macintosh változata, és most dolgoznak az 5.0 „macesítésén". A StarOffice 5.0 Sun Solaris operációs rendszeren működő változata ugyancsak ingyenesen elérhető, mégsem tettük fel a CD-re, mert Magyarországon a Solarist nagyon kevesen használják a vállalati szférán kívül.

Support

Tapasztalataink szerint a legtöbb embernek a regisztrálás okozott gondot, részben a procedúra szokatlansága miatt, részben mert sok helyen hiányoznak olyan technikai feltételek (Internet hozzáférés, fax stb.), melyek egyszerűbbé teszik az egészet. Az angol nyelvtudás hiánya egyeseknek szintén megnehezítette az eligazodást. A fenti cikkben leírtakra támaszkodva remélhetőleg most már mindenki túljut az installálás és regisztrálás akadályain. A későbbiekben igyekszünk többnyire a szoftverhasználat gyakorlati kérdéseivel foglalkozni, hogy e sokoldalú és hatékony munkaeszköz képességeinek ne csak a töredékét tudják kihasználni.

Faklen Pál

Red Hat 5.2 — kezdőknek és haladóknak

Hogyan telepítsünk Linuxot?

Az Új Alaplap előző (1999. májusi) számának CD-mellékletén örömmel fedeztem fel a Linux Red Hat disztribúciójának 5.2-es változatát, és mert nem találtam mellette részletes telepítési útmutatót, rögtön visszaemlékeztem arra, milyen helyzetben voltam, amikor először kellett Linuxot installálnom. Aki még nem csinálta, annak kicsit riasztó lehet, miket kell elkövetni ahhoz, hogy a rendszer felmásszon a gépre. Újabban ismét érdekelni kezdett a Linux, ezért elhatároztam, hogy saját tapasztalataimat az Új Alaplap olvasóival megosztva igyekszem segíteni azoknak, akik most határozták el, hogy megpróbálkoznak a Linux telepítésével. A konkrét példa a Rad Hat 5.2.

A kezdet

Akinek nem olyan az alaplapja, hogy a CD-ről bootolhat (nekem egyébként ilyen), annak a teljesség kedvéért leírom, hogyan készítse el az ún. behúzólemezeket.

A Red Hat könyvtárban van egy IMAGES könyvtár, abban találhatók az image-fájlok, ezeket kell a rawrite-tal egy formattált floppyra kiírni. Ezt megtehetjük úgy, hogy kiadjuk a

```
rawrite x:\fokusz\red_hat\images\boot.img
```

parancsot, amire a program megkérdezi, hogy hová írja. Válasz: a:. Ezután kiíródik a fenti fájl a lemezre. Jómagam egyszerűbb utat választottam: a merevlemez egyik könyvtárába átmásoltam a rawrite.exe fájlt és mindegyik img kiterjesztésű fájlt — boot.img, supp.img, rescue.img — és utána adtam ki a rawrite boot.img parancsot. Ezután már csak az a: meghajtót kérte a program. Célszerű a fenti módszerek valamelyikével elkészíteni a másik két image-fájlt tartalmazó lemezt is, mert azokra vész esetén szükségünk lehet. Menet közben még egy bootlemez készítését is felajánlja a telepítő, ez elől sem érdemes kitérni. Így összesen négy lemezre van szükségünk.

A telepítés

Azzal a lemezzel indítsuk újra a rendszert, amelyik a boot.img-t tartalmazza. Mi történik ekkor? Rövid töltögetés után egy egyszerű képernyő fogad minket, ahol három lehetőség közül választhatunk:

1. Telepítés kezdőknek.
2. Telepítés haladóknak, egyedi beállításokkal.
3. Rendszerkarbantartás. (Rescue.)

Természetesen az első lehetőséggel kell élnünk, ezt egyszerűen az Enter leütésével tehetjük meg. Tudnunk kell, hogy ezt az operációs rendszert nemcsak az egyszerű felhasználóknak találták ki, vannak benne bonyolultabb dolgok is, például léteznek a munkánkat megkönnyítő *virtuális konzolok*. Ezeket az Alt+F2 ... F5 gombokkal érhetjük el, míg az, amit kezdettől fogva látunk, az Alt+F1-en fut. A többi konzolon a következők vannak.

Alt+F2: Shell prompt (később erről még lesz szó).

Alt+F3: Telepítési napló.

Alt+F4: A kernel üzenetei.

Alt+F5: Egyéb üzenetek.

Ezek a képernyők nemcsak a telepítés során léteznek, hanem a már működő rendszerben is. Ha például meg kell oldanunk a gépen valamit, de aktuális munkánkat sem akarjuk megszakítani, egyszerűen átváltunk egy másik virtuális konzolra, ahová belépve azonnal dolgozhatunk.

Ha lenyomtuk az Entert, először azt kell eldöntenünk, hogy *install*áljuk az új rendszert, vagy *upgrade*-eljük az előző verziót. Ekkor először a forrást kérdezi meg a telepítő, illetve azt, hogy van-e SCSI eszközünk (és fel is ajánlja a helyes választ). Különbség az 5.0-s verzióhoz képest, hogy a telepítés nyelvét is kiválaszthatjuk. (Nagy előrelépés az egy évvel ezelőttihez képest, miként az is, hogy három verziószámot ugrott, és sebességben, megbízhatóságban felülmúlja az 5.0-t.) Sajnos a magyar nyelv még nem került fel a listára, de a német már igen. (Ez segítség lehet azoknak, akik a némettel jobban megbirkóznak, mint az angollal.) A billentyűzetet is ekkor lehet kiválasztani.

Az előbbieknél komolyabb első feladat a particionálás. Mindenképpen javaslom *telepítés előtt* merevlemezünk legalább két részre osztását valamelyik particionáló programmal (például fdisk). Mintegy 500 MB szabad területet — vagy többet — alakítsunk ki a lemezen. (Üres merevlemez esetén erre persze nincs szükség.) Ha már van üres — akár formázott, akár formázatlan — helyünk, akkor a *Diskdruid* nevű program segít beállítani a megfelelő partíciókat. Az első IDE portra csatlakozó master eszköz a *hda* (primary master). Ennek különböző partíciói a *hda1*, *hda2* stb. A primary slave a *hdb*, a secondary master a *hdc* és a secondary slave a *hdd*. Az SCSI eszközök az *sd* előtaggal kezdődnek és az *sda, sdb* stb. neveket veszik fel.

A Diskdruid könnyen kezelhető program. Az *Add* paranccsal készíthetünk új partíciókat. Az *Edit* paranccsal a meglévőket módosíthatjuk. A *Remove*-val eltávolíthatjuk a nem kívánt részeket, a többi funkció (*Ok, Back, Cancel*) neve is elég közismert. Itt merevlemezünkről minden információt látunk: milyen partíciók vannak és azok mekkorák, milyen típusú a fájlrendszer stb. *Vigyázzunk, hogy a hda1-et ne töröljük* — hacsak nem áll kifejezetten szándékunkban —, mert másik operációs rendszerünk esetleg még azon van (DOS, Windows, OS/2 stb.). Ha nem tettünk meg, akkor most átméretezhetjük ezt a partíciót (jelzem, hogy én még nem próbáltam ki, nem tudom, milyen következményekkel járhat).

Ha már van üres helyünk, akkor az Add gombbal hozhatjuk létre az ún. *root* partíciót (*linux native*), itt megnevezésnek csak egy / jelet kell tennünk, és a méretet beírnunk (például ha 500 MB az üres hely, én 450 MB-ot javaslok). A fennmaradó helyen (az előző példa után kb. 50 MB) lesz a lemezen a *swap* terület (a memóriabővítést szolgáló „raklap"). Ennek nagyságára az okosok a fizikai memória kétszeresét javasolják, de olvastam már olyat is,

hogy annak háromszorosával állítólag gyorsabb a rendszer. (Még egy tapasztalat: ha pontosan 500 MB a szabad terület és a natív partíciót 450 MB-ra méreteztük, akkor nem tudunk 50 MB-os swapot létrehozni, csak 48 MB-osat, mert 50 MB jelölése esetén a program olyan hibaüzenetet ad, hogy nincs elég hely a merevlemezen, ugyanis az *ext2* fájlrendszernek is kell hely.) Szóval a *swap* létrehozásakor itt nem kell semmilyen címkét adnunk a területnek, hanem csak a jobb oldalon lévő megnevezések közül ki kell választani a *swap* opciót.

Ha ezzel megvagyunk, javaslom felírni a partíciók neveit, például:

hda1 — meglévő rendszerünk
hda5 — linux natív
hda6 — swap terület

Mentés és kilépés után a telepítő megkérdezi, hogy melyik terület legyen a swap (ezt oda is írja, például a hda6), leformázza-e azt, illetve ellenőrizze-e a hibákat. Jelöljük ki a formázást (jómagam nem kértem a hibák ellenőrzését, mert az 5.0-s verzió ilyenkor hajlandó volt lefagyni). A következő kérdés, hogy hová telepítse a rendszert (hda5). Itt is választható az előző két opció.

A csomagok kiválasztása.

A Red Hat *rpm* kiterjesztésű csomagokat telepít fel, ezek az RPMS könyvtárban találhatók. A CD-n lévő verzió a szűkös hely miatt enyhén le van butítva, ezért nem célszerű a teljes (all) opciót választani, mert akkor a telepítő gyakran megáll.

Mindenképpen javaslom telepíteni például a *printtools* és a *network* eszközöket, akik pedig programoznak, azoknak a C development-et is. Egyébként ez csak a csomag kiválasztása, emellett egyesével is kiválaszthatók az egyes fájlok, melyeket szeretnénk telepíteni. Mindenképpen célszerű feltenni a felajánlott csomagokat, nem pedig egyenként kiválasztani a kívánt részeket, mert (tévedni emberi dolog) olyan egységeket is kifelejthetünk, amelyek nélkül egyes programok nem működnek.

Ezután kezdődik a fájlok tényleges másolása, ami a géptől függően *10-20 percig is (!) eltart*. A telepítésről egyébként folyamatosan kapunk információkat: látjuk, hogy melyik csomagot telepíti a program, mennyi a még hátra lévő idő, mennyi az összes szükséges idő. Ha netán olyan csomagot vagy programot jelöltünk ki, amely nincs a CD-n, arról jelzést kapunk és egy Enter után megy tovább a másolás. (Ez is előrelépés a korábbi verzióhoz képest, hiszen annál a változatnál nekem rögtön lefagyott a gépem, ha ilyen előfordult.)

Az egér konfigurálása

Kétgombos egerünkhöz mindenképpen a *háromgombos emulációt* kell választanunk. *Vigyázat!* Sok olyan egér van, amely látszólag háromgombos, de abból valójában csak kettő működik. Ezt még telepítés előtt célszerű tesztelni, mert ha nem jól választunk, akkor *X Window* alatt nem fog menni az egér, anélkül pedig ott elég körülményes dolgozni. Persze, ha már elrontottuk a beállítást, akkor se essünk kétségbe. Az X Windowból *Ctrl+Alt+Backspace* a kiléptető, és a *mouseconfig* paranccsal újra beállíthatjuk az egeret.

Képernyőbeállítás

A telepítő felismeri a PCI buszos kártyákat, ezzel nincs sok dolgunk. A monitor kiválasztása már körülményesebb. Ha monitorunk nem szerepel a felkínált listában, akkor válasszuk a *custom* típust, és utána tovább finomíthatjuk a beállítást. Ha ismerjük monitorunk képességeit, akkor válasszuk a számunkra legmegfelelőbbet (SVGA 1024x768, 70 Hz vagy jobb). Ha nem vagyunk biztosak monitorunk paramétereiben, akkor kérjük inkább a biztonságos 640x480-as felbontást, 50-60 Hz-cel. (Ezt is tudjuk később módosítani az *Xconfigurator*ral.) A kiválasztás után a *Superprobe* háromszor felvillantja a képernyőt, ekkor a színmélységen változtathatunk, vagy meghagyhatjuk a felajánlottat.

Hálózat, időzóna...

A *hálózat* beállítása csak ezután jöhet. Akinek otthon nincs helyi hálózata (LAN), az ilyen telepítését ne kérjen. Akinek van, annak viszont rendelkeznie kell az ehhez szükséges adatokkal.

Következik az *időzóna* beállítása. Itt megtaláljuk Budapestet, és választhatjuk a GMT alapértelmezését is, ekkor a rendszer automatikusan ismerni fogja a nyári időszámítást.

Ezután a rendszer betöltődésekor induló szolgáltatásokat lehet kiválasztani. Ezeken a beállításokon én nem változtattam. Ha túl sokat jelölünk ki, akkor erősen leköthetjük az erőforrásokat, esetleg feleslegesen. Nem beszélve arról, hogy utólag ezeket is bármikor módosíthatjuk, és a rendszer elindulásához az eleve meglévők elegendőek. (Akinek van hangkártyája, az persze rögtön kijelölheti a *sound*-ot.)

A nyomtató konfigurálása

Három választási lehetőség van:

— Helyi. (Saját gépünkhöz csatlakozó nyomtató beállítása.)

— Hálózati. (Unixos hálózat egyik nyomtatója.)

— SMB-vel elérhető nyomtató.

Alapesetnek vegyük az elsőt, az otthon használtat. Ha a felkínált nyomtatótípusok közül egyik sem egyezik a mienkkel, akkor keressünk valami hasonlót (például IBM graphics — Epson FX típusok), vagy ne is telepítsünk nyomtatót, ezt később szintén pótolhatjuk. Ha mégis kiválasztottunk egy nyomtatókonfigurációt, akkor a program végigvezet a lépéseken: melyik porton vannak (lp0, lp1 stb.), milyen jellegű a lapkezelés stb. (Mint mindenen, ezen is lehet később változtatni.)

Most érkeztünk el az ún. *root jelszó* megadásához. A root a rendszer adminisztrátora, neki joga van mindent beállítani, felhasználókat létrehozni, törölni, programokat telepíteni stb. Rövid jelszót nem fogad el a rendszer, biztonsági okokból legalább *hat karakter* hosszúságúnak kell lennie. (És lehetőleg tartalmazzon számot vagy írásjelet is.) *A jelszót mindenképpen jegyezzük fel*, mert enélkül nem tudjuk elindítani rendszerünket. A guruk szerint a már működő rendszerben érdemes létrehozni egy *második felhasználót*, saját nevünkre vagy kitalált névre az *adduser* paranccsal, vagy X alól a Control Panelből. Ezzel sok kellemetlenségtől óvhatjuk meg magunkat (például a letörölt fájlok sehogy sem nyerhetők vissza, bár úgy tudom, hogy ennek megoldására vannak próbálkozások).

LILO

Az utolsó lépés a LILO konfigurálása (*Linux Loader*). Két választásunk van:

— Betehetjük a LILO-t a master boot recordba (MBR), ekkor bootmanagerként fog viselkedni.

— Betehetjük a LILO-t a linuxos partíció elejére, és így csak akkor indul el a Linux, ha ezt a partíciót tesszük aktívvá.

Szerintem sokkal egyszerűbb az elsőt választani, mivel a LILO felajánlja már meglévő más partíciók felvételét, így ezeket is tudjuk majd bootolni. Ha ezt választjuk, akkor célszerű (ha volt ilyen) a másik DOS vagy Windows partíció címkéjét dos-ról (ezt adja a LILO) d-re változtatni, mert ha ezt a rendszert akarjuk indítani, akkor gyorsabb és kevesebb melléütéssel jár egy d-t begépelni, mint például azt, hogy

```
a dos inditasa
```

Tudnunk kell azt is, hogy ha a LILO az MBR-ben van, akkor minden indí-

tásnál és újraindításnál 5-6 másodpercünk van beírni egy másik bootolandó rendszer címkéjét (például: d), és ha a ez a kivárás letelik, akkor automatikusan indul a Linux.

Rendszerindítás

Miután valamennyi beállítással végeztünk, jöhet a rendszer újraindítása. Ha nincs türelmünk kivárni a gépünkön esetleg meglévő más rendszereknek bootolási lehetőséget biztosító 5-6 másodpercet, akkor a LILO felirat megjelenése után üssünk Entert.

Rövid töltögetés után megjelenik a

```
Login hostname:
```

Ekkor még csak mint root tudunk belépni, tehát írjuk be: root, majd Enter után adjuk meg a jelszót. (Azok kedvéért írtam le ilyen részletesen, akik esetleg még nem dolgoztak hálózatban.)

Ha a jelszót helyesen adtuk meg, akkor előbukkan a *shell* prompt, és innen kezdve dolgozhatunk. De mit?

Például állítsuk be a hangkártyát! Erre szolgál a *sndconfig* parancs. Automatikusan mindent beállít, majd ellenőrzésképpen megszólal Linus Torvalds, és üdvözöl minket.

A *Midnight Commander* (mc) megjelenésében hasonló a Norton Commanderhez, de kicsit eltér attól. Az *mc* paranccsal indul, első elindításakor létrehoz egy konfigurációs fájlt saját könyvtárunkban, ezt Enterrel nyugtázhatjuk.

A Linux kezelésére is igaz az a megállapítás, hogy *„ha nem törölsz, akkor igazán nagy kárt nem csinálsz"*. Tudni kell persze, hogy a Unix és Linux rendszerek (sokan a hasonlóság miatt ezeket egynek tekintik), meglehetősen diszkrét módon működnek, nem szólnak minden végrehajtott parancs után, hanem csak akkor jeleznek vissza, ha problémát észlelnek.

Azt is meg kell szokni, hogy nem a fordított törtvonal (\) a választójel az útvonal megadásában, hanem az Internethez hasonlóan a normál törtvonal (/).

Még egy lényeges különbség a DOS/Windows környezethez szokottaknak: *a Unixban nincsenek „meghajtók", hanem itt minden fájlként van definiálva*. És akkor hol találjuk meg például a CD-t? Az *mnt/cdrom* könyvtárban. Amikor azonban a DOS-hoz szokott felhasználó abba belenéz, nem lát mást, csak két pontot. Igen, mert a unixos rendszereknél nem úgy van, hogy bárki bármit feltehet a rendszerre, hiszen abból káosz és rendszerösszeomlások sorozata lenne. Az eszközöket előbb a rendszerhez kell kapcsolni a *mount* paranccsal, például:

```
mount mnt/cdrom
```

Ezután a bent lévő CD-t már ki sem lehet venni addig, amíg az eszköz nincs leválasztva (nem éppen elegáns zsargonban: unmountolva) például:

```
umount mnt/cdrom
```

Próbaképpen mountoljuk a CD-meghajtót! Most már látszik a meghajtóban lévő lemez tartalma. A Midnight Commanderrel elég egyszerű a csomagok telepítése vagy frissítése. Ha megkeresünk egy rpm fájlt (RedHat/RPMS), ennek tartalmát úgy nézhetjük meg (mivel ez egy becsomagolt fájl), hogy Entert ütünk rá. Itt legtöbbször találunk egy *INSTALL és egy *UPGRADE fájlt. Ha ezek egyikét elindítjuk, akkor (ha ez lehetséges — de erről majd később) megkezdődik a telepítés vagy az upgrade.

Vadkerti László
garp@freemail.c3.hu

StarOffice 5.0

Magas színvonal alacsony áron

StarOffice 5.0 Personal Edition Deluxe

Magas színvonalú irodai alkalmazáskészlet személyes használatra, nagyon előnyös áron.

- **Telepítés CD-ROM-ról**
- **Felhasználói kézikönyv**
- **Kiegészítő elemek**

39,95 US dollár
(+ adó és szállítási díj)
Online megrendelés:
www.stardivision.com/shop

StarOffice 5.0 Professional Edition

Irodai alkalmazáscsomag az üzleti tevékenységben történő felhasználáshoz

- **Telepítés CD-ROM-ról**
- **Felhasználói kézikönyv**
- **Kiegészítő sablonok,**
- **Mintaállományok**
- **Grafikák (clipart)**
- **Különleges betűkészletek**

169 US dollár
(+ adó és szállítási díj)
Online megrendelés:
www.stardivision.com/shop

További információk:
www.stardivision.com/shop



Az igazi alternatíva

A StarOffice értékes irodai munkaeszköz, amely minden elterjedt számítógépes környezetben használható, például Windows 95/98/NT, Linux,Solaris, OS/2 operációs rendszer alatt. A csomag a leggyakoribb irodai alkalmazások teljesen integrált készlete. Kezelõi felülete ötletes és egyszerû, amelyen azok, akik már használtak hasonló szoftvereket, szinte azonnal hatékonyan tudnak dolgozni. A sokféle dokumentumformátum kezelése révén jól együttmûködik a legelterjedtebb irodai termékekkel, kezelni tudja azok állományait, és a StarOffice-ból is átvihetõk az adatok, fájlok más rendszerekbe.

Legyen a partnerünk!

A Star Division további együttmûködésekre törekszik. Ha a Star Division termékeit forgalmazni szeretné, vagy azokkal kapcsolatos szolgáltatást, oktatási tevékenységet végezne, bekapcsolódhat a Star Division Partner Programba. Akit érdekel ez a lehetõség, keresse a kapcsolatot:
partner@stardivision.com

A StarOffice 5.0 tartalma:

StarDesktop: a munkakörnyezet

StarWriter: a szövegszerkesztő

StarCalc: a táblázatkezelő

StarDraw: a rajzolóprogram

StarImpress: a bemutatókészítő

StarBase: az adatbáziskezelő

StarSchedule: az eseménytervező

StarMail: az e-mail program

StarDiscussion: a hírolvasó

Tools: a példatár

Mindent egy helyen

STARDIVISION

Star Division Corp.
6515 Dumbarton Circle,
Fremont, CA 94555, USA

INFORMÁCIÓKÉRÉS: 24 ◄

Előzésben az IBM DB2 UDB

„Mert mi is ezt használjuk"

Tavalyi licenceladásaival az IBM előnyt szerzett az adatbázisrendszerek piacán. Ez egyrészt annak köszönhető, hogy megerősödött a DB2 értékesítése a mainframe rendszerekre, másrészt hogy a DB2 gyorsan terjed Unix és NT platformon is. Az IBM és az Oracle az utóbbi pár évben fej-fej mellett haladt az adatbázispiacon, de 1998-ban az IBM 28,9%-ról 32,3%-ra növelte részesedését, míg az Oracle lényegében azonos szinten maradt, aránya 29,4%-ról 29,3%-ra módosult. (Forrás: Reseller News)

A DB2 már régóta stabil és rugalmas adatbázisrendszernek számít, „izmos" védelemmel az adatok biztonsága szempontjából. Annak idején a nagyhírű hollywoodi filmrendező, George Lucas virgonc kis robotja is ezzel dolgozott (R2DB2). ;-) Manapság a DB2 UDB egyik legfontosabb előnye, hogy számos platform befogadta: Windows 3.x/95/98/NT, Macintosh, OS/2, Unix. Nemsokára lesz ingyenes Linux verzió is.

A DB2 egyik nagy vonzereje, hogy jelenleg szinte minden platformon ez az adatbázisrendszer nyújtja a legjobb teljesítményt. Emellett komoly és megbízható webtámogatással rendelkezik, ami a manapság terjedőben lévő elektronikus üzleti tevékenységekhez elengedhetetlen, a Java alapú meghajtók például már az alapcsomagban benne vannak, nem kell azokat a honlapokról bővítésként levadászni.

A DB2 meghajtói széleskörű támogatást nyújtanak, nemcsak a támadható felületet képező Open Database Connectivity csatornát kezelik, hanem lehetőség van a kiszolgáló által ajánlott porton keresztül is elérni az adatokat. (Ekkor még DB2 ügyfélprogramra sincs szükség, amennyiben igazolni tudjuk magunkat a DB2 „adatvédelmi szolgálata" előtt.)

A mai „builderes" fejlesztői környezeteken felnevelkedett ifjú titánoknak jelentős segítséget nyújtanak azok az eszközök, amelyek (szinte az egész rendszert átfogva) könnyebbé és egyszerűbbé teszik az adatbázisrendszer konfigurálását és adminisztrációját.

Control Center (vezérlőközpont)

A grafikus kezelőfelületnek tulajdonképpen ez a fő része. Itt a legtöbb adminisztrációs tevékenységet elvégezhetjük, de az egész rendszert nem öleli át. Tartalmaz olyan részegységeket, amelyek könnyebbé teszik a rendszer kezelését, mint például Backup Database, Create Database, Create Tablespace, Performance Configuration, Restore Database és a SmartGuide-ok. (Windowsos szóhasználatban wizardok). Alkalmazza a megszokott lebegő menü funkciót (a jobb egérgombot kell használni), amelynél az adott helyzetben elvégezhető összes metódus a szemünk elé tárul.

Client Configuration Assistant (ügyfél-segédeszköz)

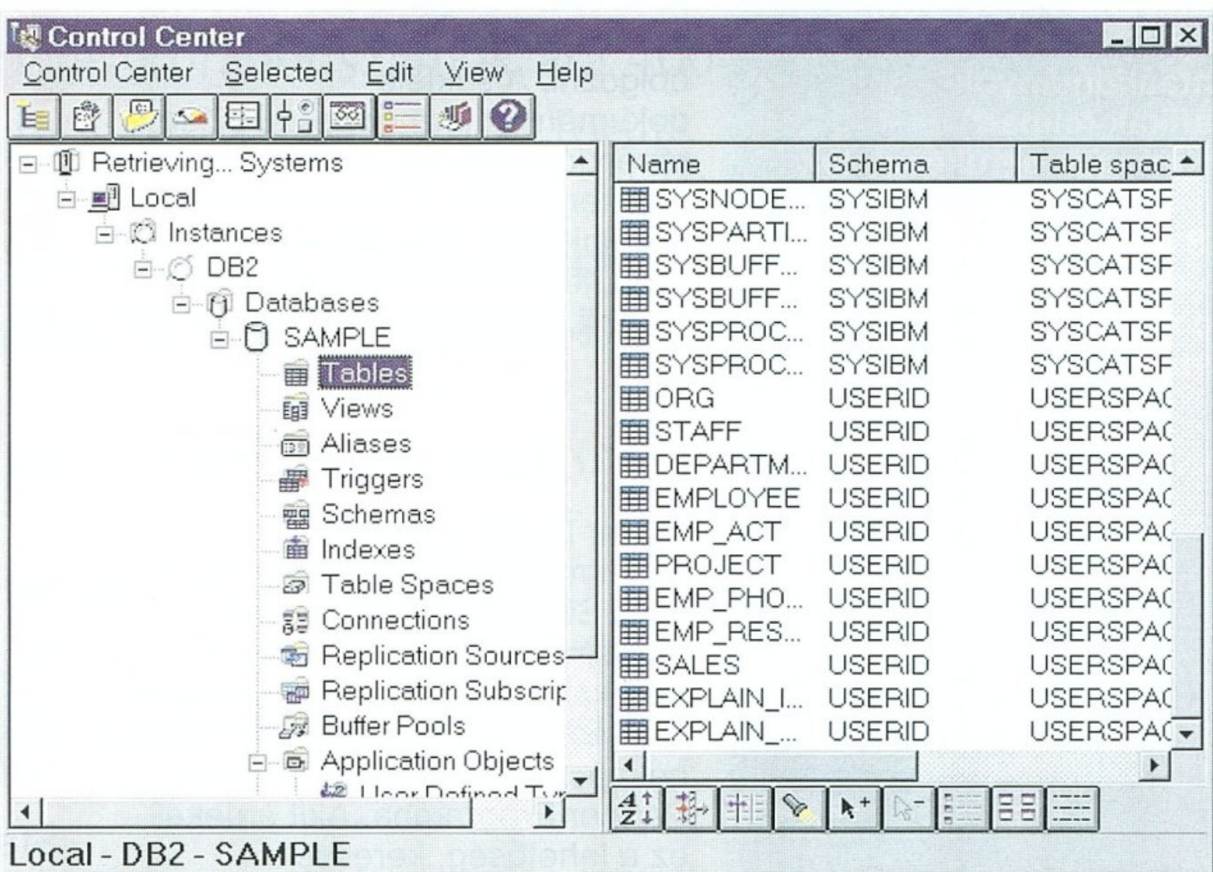
Azon adatbázisok listáját tartalmazza, amelyeket alkalmazásainkból elérhetünk. Lehetőségünk van az adatbázisokat katalogizálni (lokális vagy távoli szinten), törölni, az adatbáziskapcsolat paramétereit módosítani, a kapcsolatot tesztelni, vagy akár DB2 segédprogramot és felhasználói alkalmazást bejegyezni (bind).

Performance Monitor (teljesítményfigyelő)

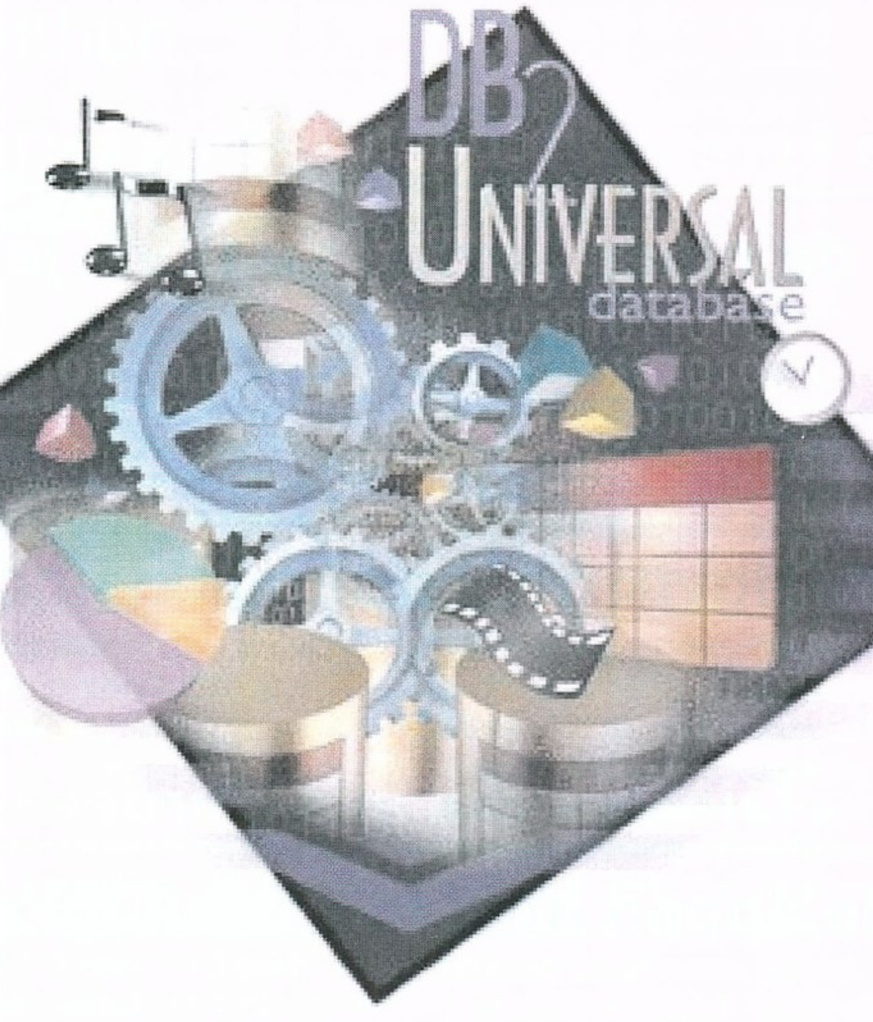
Volt, aki azt állította, hogy a DB2 nem alkalmas a rendszer monitorozására. Nem tudom, honnan szerezte információját, de egy jól konfigurált monitorozással nemcsak azt tudom megmondani, hogy a felhasználó „mikor vakarta meg a fülét, hanem azt is, hogy melyik kézzel". Ha monitorozni szeretnénk, a következő típusú eseményeket figyelhetjük meg: adatbázis, tábla, táblatér, holtpontok, kapcsolatok, tranzakciók, utasítások. Az utóbbi három esetében megadhatunk bizonyos szűrőket, ha nem akarjuk hamar teleírni merevlemezünket. Az alapértelmezett méret 200 K (4 K-s blokkokban), ha egy fájl betelt, akkor új jön létre. Megadhatunk maximális méretet vagy darabszámot az eseményállományokra vonatkozóan.

Command Center (parancsközpont)

A DB2-nek saját Command Line Interface-e (parancssori kezelése) van, amelynek azonban megvannak a hátrányai. Ha nem az operációs rendszer CLI-jéből hívjuk meg, a parancs elé



gépelt db2 karakterszekvenciával, akkor egyszerre csak egy sort tudunk bevinni, emiatt nem ő emlékszik jobban a korábbi utasításokra. Ez a Unixban kiküszöbölhető bizonyos vi funkciókkal, de az igazi megoldás: használjunk Command Centert. Nemcsak interaktív módon dolgozhatunk vele, hanem ami még fontosabb, scripteket futtathatunk benne. (A scripteket nemcsak itt futtathatjuk, hanem a Script Centerben is, és ott elő is állíthatjuk őket.)



Alert Center és Journal (őrközpont és napló)

A DB2 által generált alertek (figyelmeztetések) az Alert Centeren keresztül érhetők el. Ezek az alertek olyankor jönnek létre, amikor például „snapshot" monitort használunk és az valamilyen hibára bukkan, vagy ha egy katalogizált távoli node (csomópont) leáll, illetve más okból megszakad vele a kapcsolat. A Journalban kaphatunk információt a feladatok (job-ok) állapotáról, illetve itt tallózhatunk a három legfontosabb naplóállományban: history_log, alerts_log, messages_log.

A Universal Database Edition (UDB) lényegében a DB2 új verziója (5.x), amely hatalmasat ugrott előre az előző verziókhoz képest. Kezeli a szimmetrikus többproccesszoros rendszereket (SMP) a skálázhatóság és a teljesítmény növelése érdekében. A fontos és időigényes folyamatok végrehajtása lényegesen felgyorsult. (Lásd: LOAD, BACKUP, RESTORE stb.) De nemcsak a többprocesszoros rendszereken javult a rendszer, hanem általánosságban is jobb lett mind a teljesítmény, mind a kapacitás- és memóriakihasználás.

Új grafikus eszközöket tartalmaz, amelyek könnyebbé teszik a DB2-es rendszer telepítését, konfigurálását és adminisztrációját. Részletesebb információért sem kell a szomszédba menni, ha valaki bővebb tájékoztatásra vágyik, javaslom a DB2 saját Information Centerét, amely lényeges segítséget nyújt minden fontos témában.

Az információk védelme érdekében támogatást tartalmaz az ún. elosztott számítási környezet (Distributed Computing Environment, DCE) részére. A DCE architektúrával könnyebb a felhasználók, a jelszavak, a csoportok adminisztrációja, és maga a bejelentkeztetés is sokkal biztonságosabb.

Továbbfejlesztették a host adatbázisokkal történő kommunikációt a DB2 Connect (beépített termék) és az elosztott relációs adatbázis-architektúra segítségével (Distributed Relational Database Architecture, DRDA).

DB2 fejlesztés

A számos kapcsolódási lehetőség közül a legdivatosabbak: Web, Java, JDBC. Széleskörű multimédia támogatás, különböző bővítőkön keresztül is, amelyekkel megoldható a kiterjesztett szövegek, képek, hang- és videoanyagok tárolása az adatbázisban. Lehetővé vált a nagy objektumok tárolása is (LOB), és azokban egyaránt lehet karakteres és bináris információ (CLOB, BLOB). A LOB lehetővé teszi a keresést (tárolt képek esetén például indexelés nélkül kereshetünk színtónusokra), emellett kezeli a felhasználó által definiált típusokat (User Defined Type, UDT). Az eseménykezelés szintjén is számos feladatot megoldhatunk, például a megszokott módon indíthatunk funkciókat események hatására. Lehetőség nyílik a felhasználó által definiált és beépített függvények (SUBSTR, BLOBLENGTH stb.) használatára, illetve rekurzív SQL műveletekre.

Ma fontos szempont, hogy a Java alkalmazásfejlesztők is megtalálják mindazokat az osztályokat és meghajtókat, amelyek használatával alkalmazásaikból elérhetik a kívánt adatokat.

Kétféle elérési mód létezik a kapcsolat felépítésére. Az egyik a COM.ibm .db2.jdbc.app osztályon keresztül, a helyi hostra telepített Client Application Enabler segítségével (a CAE tulajdonképpen a DB2 kiszolgáló ügyfélprogramja), a másik a COM.ibm.db2.jdbc .net osztály, amely a kiszolgálón erre a célra kiajánlott porton keresztül éri el az adatbázist. Webes felületen is lehetőségünk van nagyon gyors adatlehívásra, főleg, ha relációs adatbázisrendszerekre szabott alkalmazásokat használunk, amilyen a Net.Data, a WebSphere kiszolgáló stb. Ezeknek az alkalmazásoknak fő erénye, hogy az esetenként lassú Java appleteket nem feltétlenül kell futtani az ügyfél oldalon, hanem azokat a kiszolgáló oldalon servletként indítjuk, és az ügyfél gép már csak HTML kódokat kap dinamikusan.

Összességében a DB2 UDB verziója (5.x) nagyon stabil, biztonságos, gyors és könnyen kezelhető adatbázisrendszer. Az UDB-nek mindenképpen hasznára vált, hogy megalkotásakor az IBM nem a „csináljuk meg gyorsan, csicsázzuk ki és adjuk el minél hamarabb" stratégiát követte, hanem a „gondoljuk végig, milyen terméket alkotunk, mert mi is ezt használjuk" jegyében fejlesztette ki ezt a verziót.

Valóban komoly biztonsági és védelmi előírásoknak eleget tevő rendszereket építhetünk ki vele, a leghatékonyabb tranzakciókezeléssel, amit csak tovább erősít a magas szintű rendszeradminisztráció.

Schultz Péter
pschultz@hu.ibm.com

További információk
IBM DB2 honlap:
http://www.software.ibm.com/data/db2
IBM technikai dokumentációk:
http://www.redbooks.ibm.com
DB2 Magazine:
http://www.software.ibm.com/data/db2/products

Helyzetkép külföldön és itthon

Az Internet számokban

Egyre több cég arra a feltételezésre alapozza üzleti stratégiáját, hogy a hálózat használata a jövő század elején ugyanúgy hozzá fog tartozni mindennapjainkhoz, mint ma a telefonálás. Az internetezőkre vonatkozó adatok egyelőre azonban elég bizonytalanok, a weblapok áradatát a legnagyobb keresőmotorok sem képesek követni, és nehéz reális képet alkotni az Internet használatának módjáról is. Úgy látszik, hogy a decentralizáltságnak ez az ára. A legutóbbi felmérések alapján mégis igyekszünk hiteles áttekintést adni az Internet elterjedtségéről szerte a világon — és itt Magyarországon.

Az Intel előrejelzése szerint a Y2K-probléma mellett a 2000. év azért is fordulópont lesz a számítástechnika történetében, mert átlépünk egy bűvös határt: több személyi számítógépet fognak eladni a világon, mint ahány televíziót. A PC így hamarosan tényleg ugyanolyan széles körben használt tárggyá válik majd, mint a tévékészülék. A tendenciát egyfelől megerősíteni látszik, hogy Ausztráliában, Kanadában, Dániában és Koreában már tavaly is kelendőbb volt a személyi számítógép a televíziónál, és az is igaz, hogy nemcsak az olcsó, márka nélküli gépek kategóriájában várható valóságos árzuhanás, hanem például a laptopok körében is.

Eközben azt sem szabad elfelejtenünk, hogy a csökkenő árak az eddigi számítógéphasználóknak kedveznek leginkább, egyelőre még nem várható a „szűz tömegek" beáramlása. A kimutatások szerint ugyanis a vevők mintegy 80%-a régebbi számítógépét cseréli le. Még az Amerikai Egyesült Államok háztartásainak is csupán 50%-ában van számítógép, világviszonylatban pedig a háztartások alig 20%-ában. Ez a fennmaradó 80% túlságosan sok ahhoz, hogy egy-két, vagy akár néhány év alatt „telíteni" lehessen a piacot.

„Kerekítési hiba"

Az Internet fejlődése mindamellett továbbra is rendkívül gyors. A Computer Industry Almanac becslése szerint 1996 és 1998 között 61 millióról 147 millióra nőtt azok száma, akik hetente legalább egyszer használják az Internetet. Az előrejelzések 2000-re 320, 2005-re 720 millió felhasználót jósolnak, aminek bekövetkezése egyúttal azt is jelenti, hogy megszűnik az amerikai túlsúly. Jelenleg ugyanis a felhasználók 52%-a él az USA-ban (76,5 millió fő), és bár abszolút számuk 2005-re elérheti a 207 milliót is, részarányuk a világ megnövekedett internetező közösségében már csak 29%-ot tesz majd ki.

A népesség számához viszonyítva az internetezés aránya már ma is a skandináv országokban a legnagyobb. Export-import üzletek lebonyolítására is Skandináviában használják a legtöbben az Internetet. Az eufórikus torzítások elkerülése érdekében azonban meg kell jegyezni, hogy bármily gyakran hangzik is el manapság az e-business és az e-commerce, még a „digitális vállalati idegrendszer" prófétája, Bill Gates is elismeri, hogy „1998-ban a hálózati forgalom összesen sem volt több mint egy elhanyagolható kerekítési hiba öszege a világ teljes üzleti forgalmában: a hét legnagyobb gazdaság forgalmának mindössze 0,5 százalékát tette ki." Más mutatókat tekintve is hasonló arányokat tapasztalhatunk: a New Scientist c. tudományos folyóirat szerint a Földön már közel félmilliárd mobiltelefonos van, és ehhez képest (különösen pedig a világ össznépességéhez képest) meglehetősen alacsony az e-mailezők és a Weben böngészők száma.

Otthonról alig...

A Szonda Ipsos és a GfK Hungária 1998 második felében tízezer fős reprezentatív mintán alapuló felmérést végzett. Arra az eredményre jutottak, hogy Magyarországon összesen mintegy 800 ezren használják az Internetet. Ebből otthonról 120 ezren, munkahelyről és iskolából 550 ezren, ismerőstől és baráttól pedig további 130 ezren férnek hozzá a hálózathoz. A gyakoriságot illetően 100 ezer ember használja az Internetet naponta, további 150 ezer hetente több alkalommal, a többiek pedig ennél ritkábban.

Más felmérések azonban ettől eltérő eredményeket hoznak ki. Az Internetto 1999 áprilisi vizsgálódása szerint az 1998-as mintegy 254 ezerhez képest 1999 első negyedévében 522 ezren interneteztek. Ebben vegyesen szerepel a használat intenzitása, egyaránt benne vannak olyanok, akik hetente egyszer megnézik elektronikus postaládájukat, és olyanok, akik naponta több órát töltenek a hálózaton. Nálunk is érdemes párhuzamot vonni más elektronikus eszközök elterjedtségével, jelenleg kb. 3 millió hagyományos telefon és több mint 1 millió mobiltelefon van az országban, számítógép pedig továbbra is csak az átlagosnál magasabb jövedelműek otthonában található (számuk néhány százezer darab).

A gyors hazai internetesedésnek (a használat egy éven belüli megduplázódásának) elsődleges oka a Sulinet-program volt, melynek keretében 1410 középiskola kapcsolódott rá az Internetre. Jelentősen hozzájárult a növekedéshez az ún. akadémiai hálózat fejlesztése és az alacsonyabb éjszakai telefontarifa is.

Az Internetet használók száma néhány országban, 1999 elején

(Becslés, millió fő)

Egyesült Államok	76,50
Japán	9,75
Ausztrália	4,36
Tajvan	1,65
Kína	1,58
Európa összesen	36,02
Ebből:	
Nagy-Britannia	8,10
Németország	7,14
Franciaország	2,79
Svédország	2,58
Olaszország	2,14
Spanyolország	1,98
Hollandia	1,96
Finnország	1,57
Norvégia	1,34

A legnagyobb magyarországi Internet szolgáltatók	
Modemes szolgáltatók	**Előfizetőik száma**
Matávnet (Kék számmal, helyi tarifáért hívható).	30 000
Elender (Az 1999. márciusában átvett Compuserve előfizetőkkel együtt).	17 000
DataNet	10 000 felett
Euroweb	5000

A fentiek alapján arra lehet következtetni, hogy a növekedés hamarosan megtorpan, mert az általános iskolák — az eredeti tervektől eltérően — már nem kapnak ingyenes hozzáférést. Az 522 ezerből kb. 62 ezer az otthoni modemes kapcsolat valamelyik szolgáltatóhoz. Vagyis az Internet a túlnyomó többség számára még mindig csak a munkahelyen vagy egy oktatási intézményen keresztül hozzáférhető.

Munkahelyen, iskolában

Nézzük most azt, hogy mit lehet tudni a nagyobbik táborról. Elsősorban a Hungarnet Egyesület kutatói hálózatát (HNONE) érdemes megemlíteni, azt kb. 260 ezer ember használja. Minden fontosabb kutatóintézet és könyvtár rajta van. A felhasználók megoszlása: 200 ezer egyetemi hallgató, 10 ezer kutató, 50 ezer könyvtárlátogató.

A Sulinetet az Elender szerint mintegy 150 ezer középiskolás használja több-kevesebb rendszerességgel, a kisebb-nagyobb vállalkozásoknak köszönhetően becslések szerint tavaly mintegy 20 ezer, idén mintegy 40 ezer fő fér hozzá az Internethez a Miniszterelnöki Hivatal pedig 2500 embert tart számon az X.400-as levelezőrendszer használójaként.

Az Amerikai Egyesült Államokban az internetezők demográfiai összetétele ma már megközelíti az átlagnépességét, Magyarországon azonban még lényegesen eltér attól. 1998-ban a felhasználóknak alig 15%-a volt nő, az átlagéletkor 30 év alatt maradt (ami persze nem meglepő, hiszen a középiskolások és egyetemisták együtt kb. 350 ezren vannak).

Mindez azonban nem jelenti azt, hogy nemzetközi viszonylatban nagyon rosszul állunk. Az ír NUA felmérése szerint (amely sokak szerint persze nem igazán megbízható) az alábbi rangsor alakult ki az európai országok között az internetezőknek az összlakossághoz viszonyított arányát tekintve:

Izland, Svédország, Finnország, Dánia, Anglia, Norvégia, Svájc, Írország, Németország, Szlovákia, Hollandia, Spanyolország, Ausztria, Magyarország, Franciaország, Belgium, Olaszország, Csehország, Portugália, Lengyelország, Észtország, Törökország, Görögország, Oroszország.

Dacára a viszonylag előkelő helyezésnek, ne felejtsük el, hogy az összlakossághoz képest 6-7-szer több izlandi internetezik, mint ahány magyar, és az USA-ban már az emberek 37%-a használja az Internetet. Ráadásul az amerikai törvényhozás 1996-ban bevezette az „általános hozzáférés" fogalmát, melynek szellemében „a távközlési piac versenyét megfelelő szabályozáspolitikával úgy kell irányítani, hogy előnyeit a legtöbb fogyasztó élvezhesse" — olvasható Nyírő András és Turi László erről a témáról szóló összeállításában.

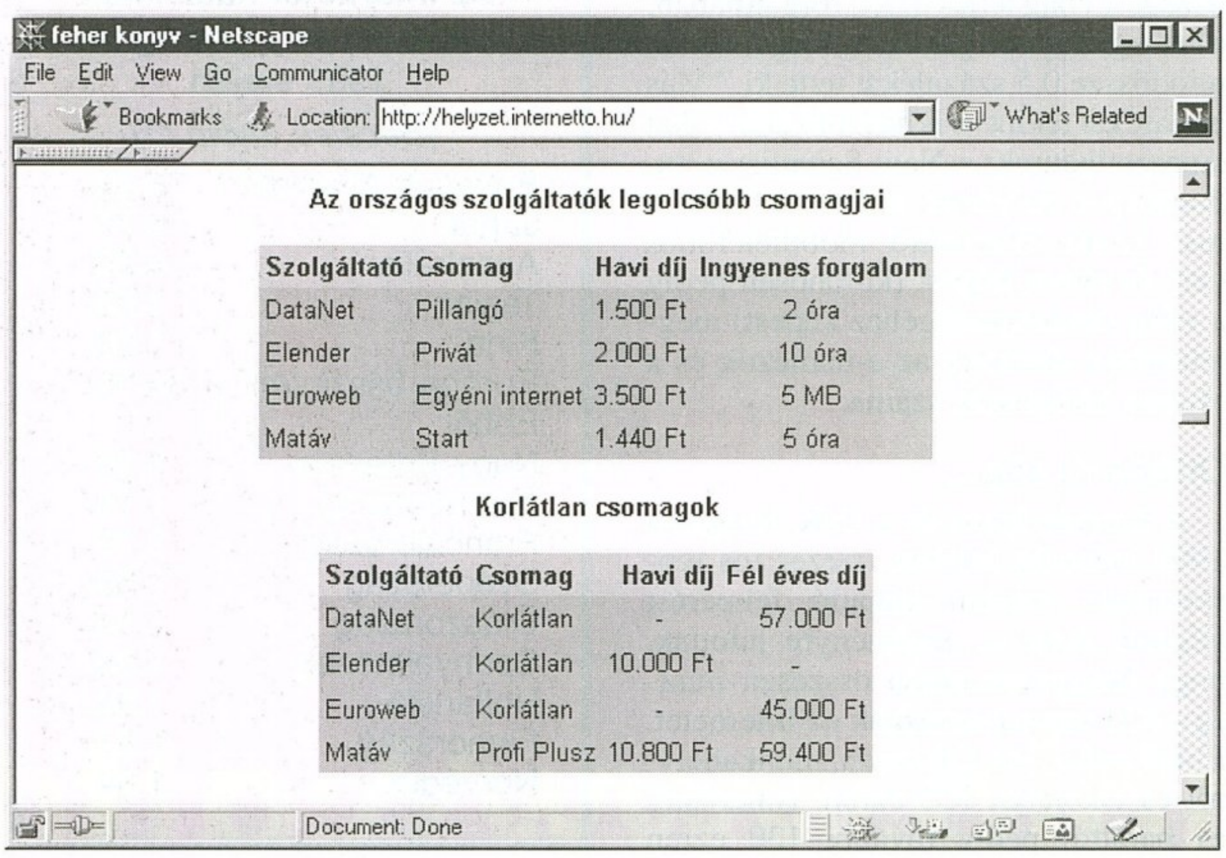
A levelezés szinte mindenkié

A mindenki számára elérhető Internetet egyelőre leginkább az ingyenes levelezőrendszerek valósítják meg Magyarországon. A Soros Alapítvány C3 központja által működtetett Freemail nem kevesebb, mint 135 ezer rendszeres használóval rendelkezik, ami 1998-hoz képest 3,5-szeres növekedést jelent, és működik már az első négy, szintén a C3 által felállított, ingyenes utcai webterminál is. A Drótpostagalamb szolgáltatást (könnyen kezelhető, magyar nyelvű levelezőrendszer, amely azonban reklámokat is eljuttat a felhasználóknak) 45 ezer ember használja (néhány hónappal az 1997 végi indulás után még csak 9500-an voltak). Megemlítendő továbbá az 1998 eleje óta létező Egon levelezőrendszer is. Az Interneten és a dial-up rendszerben egyaránt elérhető Freemaillel vagy a Drótpostagalamb rendszerrel szemben (ahol a cég által rendelkezésre bocsátott „tárcsázós" levelezőszoftver működik) az Egon a levelezést a Hotmailhez hasonlóan a webfelületen szervezte meg.

Szintén ingyenes szolgáltatás az 1998 márciusában indított Nexus, amely kombinált ingyenes e-mail-t és webhelyszolgáltatást biztosít (ma mintegy 34 ezer embernek, havonta kb. 120 új felhasználó belépésével), továbbá az Extra, amelynek szintén tízezres nagyságrendű felhasználóköre van (ez a szám havonta átlagosan 20%-kal nő).

Végezetül említsük meg, hogy bár sok szó esik az Internet reklámlehetőségeiről is, a valóság ezzel még nincs szinkronban. Magyarországon 1998-ban összesen mintegy 60 millió forintot költöttek az Interneten elhelyezett reklámokra — akkora összeget, amelyből az MTV1-en televíziós főműsoridőben negyedóránál tovább nem lehetne tévéreklámot sugározni.

Galántai Zoltán



Az országos szolgáltatók legolcsóbb csomagjai

Szolgáltató	Csomag	Havi díj	Ingyenes forgalom
DataNet	Pillangó	1.500 Ft	2 óra
Elender	Privát	2.000 Ft	10 óra
Euroweb	Egyéni internet	3.500 Ft	5 MB
Matáv	Start	1.440 Ft	5 óra

Korlátlan csomagok

Szolgáltató	Csomag	Havi díj	Fél éves díj
DataNet	Korlátlan	-	57.000 Ft
Elender	Korlátlan	10.000 Ft	-
Euroweb	Korlátlan	-	45.000 Ft
Matáv	Profi Plusz	10.800 Ft	59.400 Ft

Megjegyzés:
A Internet használatára vonatkozó nemzetközi adatok lényegében a Jövőnéző hírleveléből származnak (http://www.jovonezo.hu), a magyar felmérési eredmmények pedig az Internetto összeállításából (http://helyzet.internetto.hu).

A hónap vírusa (II. felvonás)

Még mindig a Melissa

Kissé rendhagyó módon vissza kell térnünk előző havi vírusunkra, a Melissára, amely közvetlenül áprilisi számunk lapzártája előtt bukkant fel. Gyors elemzésünk teljesen helytállónak bizonyult, de indokolt foglalkozni azzal a szinte példátlan eseménysorral is, amit ez a vírus kiváltott.
A Melissa vírus világszerte heves ellenreakciót váltott ki a vírusok ellen küzdők körében. A vírusokat eddig inkább beletörődéssel elszenvedő számítógépes közösség nem maradt tétlen, és el is érte, hogy a vírussal ilyen durva szórakozást űző ismeretlen személye ne maradjon homályban, és ne ússza meg az ügyet büntetlenül.

A Melissa vírust március 26-án délelőtt postázták az alt.sex hírcsoportra, egy szexszel foglalkozó weblapok jelszavait tartalmazó listának álcázva a skyrocket@aol.com címről. E cím gazdája később tagadta, hogy köze van a vírus postázásához, mert nevéhez és jelszavához illegálisan jutott hozzá a valódi terjesztő.

AOL + FBI

Az első postázástól számítva a vírus pár óra alatt viharos gyorsasággal terjedt el, néhány szerveren akkora levélforgalmat idézve elő, hogy azokat le kellett állítani (így járt többek között az Intel, a Microsoft és a Lucent is). A hét végéig több mint 300 cég jelzett fertőzést, és az legalább 100 ezer PC-re terjedt ki. A legkomolyabb fertőzések az Egyesült Allamokban alakultak ki.

Az FBI megfeszített erővel kutatott a tettes(ek) után, és ebben a harcban nem várt szövetségesre talált: az America Online céget. Feladva az Internetszolgáltatók általános hozzáállását, amellyel előfizetőik személyiségi jogait védik, az adatokat a végsőkig bizalmasan kezelve, az AOL az FBI rendelkezésére tudta bocsátani azt az információt, amely az eredeti üzenet feladójához elvezetett. Ez a segítőkészség annyira szokatlan volt, hogy az USA szenátusa is meghallgatta az AOL képviselőit, vajon igyekezetükben nem sértették-e meg a személyiségi jogokat.

Köztudott, hogy a szolgáltatók és a szerverek üzemeltetői megabájtszámra gyártják az eseménynaplókat, amelyek nyomán elvileg minden felhasználói tevékenység visszakereshető. Ugyanakkor az is köztudott, hogy ennek az irdatlan mennyiségű adatnak a feldolgozására csak végszükség esetén vállalkoznak. Hát most, úgy tűnik, a végszükség esete forgott fenn. Így ezen naplókat átvizsgálva eljutottak ahhoz a telefonszámhoz, amelyről modemen keresztül az ominózus dokumentum feljutott az alt.sex hírcsoportra.

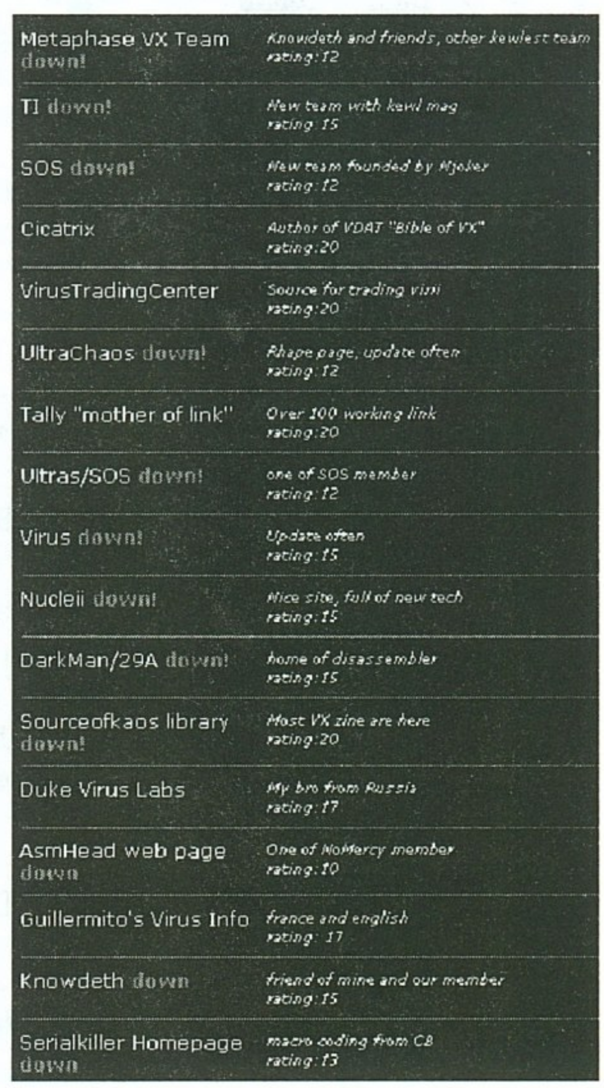
Az FBI-nak ezután már rutinmunka volt kideríteni a telefon gazdáját, egy David L. Smith névre hallgató programozót, akit a testvére házában őrizetbe is vettek. Nyakába zúdították a számítógépes visszaélés kapcsán elképzelhető összes vádat, melyek összbüntetése akár 40 évi börtön és 480 ezer dollár pénzbüntetés is lehet. A két legfontosabb vádpont az volt, hogy egyrészt a hatalmas levéltömeg miatt a vírus lehetetlenné tette a normális munkafolyamot, másrészt pedig a felhasználó tudta nélkül küldte szét a fertőzött dokumentumot, amely akár bizalmas adatokat is tartalmazhatott.

Két legyet is?

A kutatás során felmerült, hogy Smith azonos lehet a VicodinES néven hirhedtté vált vírusíróval, aki számtalan makróvírus megírásáért felelős. A nyomok arra utaltak, hogy Smith több álnevet is használt levelezése során, és hogy ezek egyike a VicodinES volt. Az mindenesetre bizonyosnak látszik, hogy az alt.sex-re került dokumentum az ő számítógépéről származik. Összevetve ugyanis a bizonyítottan Smith gépéről kikerült dokumentumokkal, kiderült, hogy a rajtuk található egyedi azonosító (a közelmúltban nagy vihart kavart, személyiségi jogokat is sértő GUID) megegyezik. Azonban, mint arra több szakértő is rámutatott, ez az azonosító csak azt mutatja meg, hogy kinek a gépén kezdték el írni a vírust tartalmazó makrót, vagy annak egy korábbi változatát.

Mivel a vírusíró rendszerint nem a nulláról indul el, hanem valamilyen korábbi vírust módosít, szinte semmit nem jelent ez az azonosság. Annyira nem, hogy más vírusírók (például az ALT+F11 névre hallgató egyén) egyes termékeiben is megtalálható volt ugyanaz a GUID. Tudva, hogy ez utóbbi volt az, aki VicodinES egyik korábbi vírusát átírta Office 2000-re, Ez egyáltalán nem meglepő. A vád szempontjából persze mellékes körülmény, hogy valóban Smith írta-e a vírust.

Az a kód alapos vizsgálatából kiderült, hogy két különböző programozási stílussal elkészített, jól különválasztható rész keveredik benne. Az egyik a szokványos vírusfunkciókért felelős, és korábbi makróvírusokban már használt elemeket tartalmaz. A másik részt, amely az Outlookon keresztül való terjedést szolgálja, a változók elnevezésének sajátosságai alapján feltehetőleg egy német anyanyelvű írta. Mindez elvileg nem zárja ki, hogy a vírust teljes



egészében egy ember fejlesztette, de sokkal valószínűbb, hogy a második részt a Melissa írója valahol készen találta, és minimális módosításokkal átemelte saját vírusába.

A nyertesek a vírusirtók

Nem hiszem, hogy a vírus írója vagy terjesztője azzal a szándékkal küldte el azt az alt.sex csoportba, hogy ekkora galibát okozzon világszerte. Inkább csak jó heccnek vélhette. Előzőleg is felbukkantak már a hírcsoportokban fertőzött dokumentumok. A balszerencse most az volt, hogy a levelezésen keresztüli terjedést megvalósító rész túlságosan effektívnek bizonyult. (Hogy miért, azt megírtuk előző számunkban.) Ez azonban nem nyújt felmentést a vádak alól. Ha gondatlanságból is, de súlyos károkat okozott. Itt az ideje, hogy egyszer végre megtanulják, nem szabad a tűzzel játszani, vagy ha mégis játékos kedvükben vannak, akkor vállalják a következményeket is.

Érdekes mellékhatásai is voltak az eseményeknek. Bár ezúttal senki nem vádolhatta a víruskeresőket azzal, hogy ők maguk idézték elő a vírus terjedését, a nagy haszonélvezők mindenképpen ők. A tőzsdén szereplő cégek (Symantec, Network Associates) részvényeinek árfolyama rohamosan emelkedni kezdett, weblapjaik forgalma többszöröse nőtt.

A vírusírók és vírusterjesztők weblapjainak jelentős része viszont a sikeres elfogás hatására ideiglenesen bezárt (lásd a „down" jelzést az 58, oldali képen), illetve egy részüknél az FBI lefoglalta a szerverként működő számítógépeket. Több hírhedt vírusíró (például a Codebreakers csoportba tartozó Opic) pedig bejelentette visszavonulását. Nem sok könnycsepp hullt értük.

A Melissa körüli hercehurca ráirányította a figyelmet az intenetes hírcsoportokba postázott vírusokra. A vírusoknak a hírcsoportokban való felbukkanása általában figyelmetlenségből, nem megfelelő elővigyázatosságból fakad. Az orosz relcom.commerce csoportokban például hemzsegtek a CAP makróvírussal fertőzött dokumentumok. A vírusok egy része viszont szándékosan, beküldőjének ártó szándékától vagy tréfás kedvétől vezéreltetve jut célba. Ez utóbbi csoport kedvelt előfordulási helyei azok a hírcsoportok (jellemzően az alt.sex), ahol nagyobb valószínűséggel fordulnak elő a számítástechnikában kevéssé jártas felhasználók, vagyis akik elég balgák ahhoz, hogy a levelekhez csatolt vírust lefuttassák.

A megoldás: őrjárat

Mit lehet tenni a rosszindulatú vírusterjesztők ellen? Van több megoldás is, csak kevés figyelmet kapnak. A Dr.Solomon's vírusfejlesztő csapat indította el például a Virus Patrol nevű szolgáltatást. Ez egy olyan automatikus program, amely naponta végignézi a hírcsoportokba küldött üzeneteket, és ha valamelyik mellékletben vírust talál (a felismerésre a legjobb víruskeresőnek tartott Dr.Solomon's AVTK-t használja), akkor ugyancsak automatikusan küld egy figyelmeztető üzenetet ugyanabba a hírcsoportba. Mielőtt tehát egy üzenet mellékletét megnyitnánk, böngésszük végig a hírcsoportot, hátha a Virus Patrol már kiszúrta a vírust.

Szappanos Gábor

A rejtjelezés rejtelmei, avagy kulcs a zárban felejtve?

Tisztelt Fehér Zsolt Úr!

Az Új Alaplap 1999. márciusi számában jelent meg Nem „a lábtörlő alatt" című válasza olvasói észrevételemre. Ebben téves állítások is szerepelnek. Vegyük sorra:

1. Idézem: *„... alapvetően azt az ötletet szerettem volna terjeszteni, hogy több rövidebb kulcsszó együttes használata sokkal nagyobb biztonságot nyújt, mint egy hosszabb kulcsszóé."* Ez tévedés.

Legyenek N, K1, K2, K3 és R nyolcbites változók. Ismeretes, hogy:

```
((N XOR K1) XOR K2) XOR K3 =
N XOR ((K1 XOR K2) XOR K3) =
N XOR Ke                              [1]
```

ahol

```
((K1 XOR K2) XOR K3) = Ke             [2]
```

valamint, ha

```
R = N XOR Ke                          [3]
```

akkor

```
N = R XOR Ke                          [4]

Ke = R XOR N                          [5]
```

és végül

```
0 XOR Ke = Ke                         [6]
```

Mit javasol Ön? Vegyünk három kulcsszót, legyen ezek jelölése: *K1*, *K2* és *K3*. A kulcsszavak hossza legyen *h1*, *h2* és *h3* bájt. A rejtjelezendő állomány bájtjait első lépésben XOR-oljuk a K1 bájtjaival folyamatosan. A kapott eredményre végezzük el ugyanezt a műveletet a *K2* bájtjaival. Befejezésül ugyanígy járjunk el *K3*-mal is.

Mit állítok én? Először XOR-oljuk a *K1*, *K2* és *K3* bájtjait folyamatosan. Eredményül egy *he* bájt hosszúságú — nevezzük *Ke*-nek — kulcsszót kapunk. (A *he* értékével kapcsolatos észrevétele egyébként helyes.) A rejtjelezendő állomány bájtjait most XOR-oljuk folyamatosan a *Ke* bájtjaival. Egy példa segítségével bebizonyítom, hogy mind a két módszerrel azonos eredményre jutunk. Legyen a rejtjelezendő szövegrészlet (az Ön szövegéből idézek): *„Termeszetesen a komoly rendszerek (DES, Blowfish stb) szinte"* (Az ékezetet az é betűről szándékosan hagytam el.) A rejtjelezendő szöveg ASCII hexakódban:

```
54 65 72 6D 65 73 7A 65 74 65 73 65 6E 20 61
20 6B 6F 6D 6F 6C 79 20 72 65 6E 64 73 7A 65
72 65 6B 20 28 44 45 53 2C 20 42 6C 6F 77 66
69 73 68 20 73 74 62 29 20 73 7A 69 6E 74 65
```

Legyenek a kulcsszavak:

```
K1="EZ "  (45 5A 20)                  h1=3
K2="EGY " (45 47 59 20)               h2=4
K3="HIBAS" (48 49 42 41 53)           h3=5
```

Rejtjelezzük a nyílt szöveget a *K1* kulcsszóval. Az eredmény:

```
11 3F 52 28 3F 53 3F 3F 54 20 29 45 2B 7A 41
65 31 4F 28 35 4C 3C 7A 52 20 34 44 36 20 45
37 3F 4B 65 72 64 00 09 0C 65 18 4C 2A 2D 46
2C 29 48 65 29 54 27 73 00 36 20 49 2B 2E 45
```

Rejtjelezzük ezt *K2*-vel, majd a kapott eredményt *K3*-mal. Azt kapjuk, hogy:

```
1C 31 49 49 29 5C 2F 5D 50 34 38 2C 2C 7C 4B
0D 3D 4A 30 46 41 32 61 33 36 3B 54 54 24 51
26 56 4C 63 78 0C 0C 0C 14 16 15 42 31 4C 50
23 39 2A 61 3D 45 4E 74 06 3C 48 45 2E 36 36
```

A második módszer szerint képezzük a *Ke* kulcsot (*he* = 3*4*5 = 60 bájt):

```
48 54 3B 24 4C 2F 55 38 24 51 4B 49 42 5C 2A
2D 56 25 5D 29 2D 4B 41 41 53 55 30 27 5E 34
54 33 27 43 50 48 49 5F 38 36 57 2E 5E 3B 36
4A 4A 42 41 4E 31 2C 5D 26 4F 32 2C 40 42 53
```

A nyílt szöveget rejtjelezzük a *Ke* kulccsal:

```
1C 31 49 49 29 5C 2F 5D 50 34 38 2C 2C 7C 4B
0D 3D 4A 30 46 41 32 61 33 36 3B 54 54 24 51
26 56 4C 63 78 0C 0C 0C 14 16 15 42 31 4C 50
23 39 2A 61 3D 45 4E 74 06 3C 48 45 2E 36 36
```

Amint az várható volt, a két eredmény azonos. Tehát több rövid kulcsszó együttes használata semmivel sem nyújt nagyobb biztonságot, mint egy hosszú kulcs alkalmazása az Ön által javasolt rejtjelezéssel. A fejtőnek ugyanis nincs szüksége a *K1*, *K2* és *K3* kulcsszavak meghatározására, elegendő csupán a *Ke* kulcsot megtalálnia. A továbbiakban ezzel fogok foglalkozni.

Újra idézek: *„Valóban kíváncsi lennék, hogy a válaszában említett ... Kasiski módszer mennyiben alkalmas arra, hogy már a titkosítási eljárás előtt is látszólag véletlenszerű bájtértékeket tartalmazó ZIP, EXE, JPG stb. állományokat visszafejtse."* Nézzük tehát ezeket a kérdéseket.

2. A végrehajtható állományok forráskódját valamilyen *programozási nyelven* írják. A programozási nyelvek, a *természetes nyelvekhez hasonlóan*, nagyon jól meghatározható statisztikai jellemzőkkel rendelkeznek. Igaz ugyan, hogy a forráskódot lefordítják az adott processzor által értelmezhető gépi kódra, de ez a kód is elemezhető statisztikailag. Az Ön által javasolt rejtjelezési eljárás megfejtéséhez azonban — ha segítségével EXE, DLL, SYS, stb. fájlokat rejtjeleznének — az esetek jelentős részében nem is lenne szükség semmilyen statisztikai elemzésre. Ha a Norton Commanderrel vagy a Norton Utilities Diskedit programjával beletekintenénk mondjuk a Windows 95 néhány DLL-jébe, akkor például az 1. ábrán bemutatott részletet láthatnánk.

A 03FF offset címtől felfelé nagyon sok DLL ugyanígy néz ki. Vannak persze olyanok, amelyekre ez nem igaz, de vannak olyanok is, amelyekben sokkal nagyobb a nullával

```
offset

002C0   00 00 00 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00   ................
        ..........................................................................
003E0   00 00 00 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00   ................
003F0   00 00 00 00 | 00 00 00 00 | 00 00 00 00 | 00 00 00 00   ................
00400   55 8B EC 6A | FF 68 71 10 | 00 10 64 A1 | 00 00 00 00   Uï.j.hq...dí....
```

1. ábra

```
56130   3C 3C 4E 4D | 53 47 3E 3E | 00 00 52 36 | 30 30 30 0D   <<NMSG>>..R6000.
56140   0A 2D 20 73 | 74 61 63 6B | 20 6F 76 65 | 72 66 6C 6F   .- stack overflo
56150   77 0D 0A 00 | 03 00 52 36 | 30 30 33 0D | 0A 2D 20 69   w.....R6003..- i
56160   6E 74 65 67 | 65 72 20 64 | 69 76 69 64 | 65 20 62 79   nteger divide by
56170   20 30 0D 0A | 00 09 00 52 | 36 30 30 39 | 0D 0A 2D 20    0.....R6009..-
56180   6E 6F 74 20 | 65 6E 6F 75 | 67 68 20 73 | 70 61 63 65   not enough space
56190   20 66 6F 72 | 20 65 6E 76 | 69 72 6F 6E | 6D 65 6E 74    for environment
561A0   0D 0A 00 FC | 00 0D 0A 00 | FF 00 72 75 | 6E 2D 74 69   ..........run-ti
561B0   6D 65 20 65 | 72 72 6F 72 | 20 00 01 00 | 52 36 30 30   me error ...R600
```

2. ábra

feltöltött terület. Az egyenlet [6] értelmében a rejtjelezés végrehajtása után ezen a területen a *Ke* kulcs részben vagy egészében, esetleg többször megismételve nyíltan meg fog jelenni. Egyes C fordítók hibaüzenetei lehetnek például a 2. ábrán bemutatotthoz hasonlóak (gyakran a végrehajtható állomány végén elhelyezve).

Ezekben az esetekben jó eséllyel alkalmazható a *valószínű szavak módszere*. Érdemes belenézni más fordítók által előállított fájlokba is. A fordítóprogramok — ha kell, ha nem — nagyon sok rutint eleve befordítanak a végrehajtható állományokba. Ismerve ezek kódját és figyelembe véve az egyenletet [5], a Ke azonnal meghatározható. (Tanulságos lehet a Word, az Excel vagy a .GIF fájlok tanulmányozása is.) A példákat vég nélkül sorolhatnám.

Az Ön eredeti cikkében még utalás sincs arra, hogy minden fájlt más-más kulcsszavakkal kellene rejtjelezni a minimális védelmi szint elérése érdekében. A *Ke*-t valamelyik rejtjelezett állományból meghatározva a támadó már nem *fejti*, hanem *megoldja* a többi rejtjelezett fájlt. Tehát állításával ellentétben a javasolt módszer *kevésbé alkalmas* a bináris fájlok védelmére, mint a szöveges állományokéra, melyeket — mint azt első észrevételemben már jeleztem — nagyon gyengén védi. Idézem: „*... a módszert ...... csak a bináris fájlok titkosítására tartom megbízhatónak.*" A fentiekből remélhetőleg egyértelműen kiderül, hogy ez az elképzelés teljesen hibás.

3. Néhány szót szeretnék mondani a tömörített fájlokról. Igen, a tömörítő eljárások hatékony segédeszközei lehetnek a nagy megbízhatóságú rejtjelező rendszereknek, amelyek előszeretettel alkalmazzák is ezeket.

A tömörítő programok jelszavas védelmi opciójának használata meglepően jó védelmet nyújthat bizonyos típusú támadásokkal szemben. (A korai verziókról ez nem igazán mondható el.) Mindenesetre lényegesen jobbat, mint az Ön által ismertetett eljárás. Tehát a javasolt rejtjelezésre ebben az esetben nincs is szükség.

4. A WordPerfect, a Quattro és a Paradox állományokra való hivatkozásáról egy közmondás jut eszembe: „Az okos más kárán (hibájából) is tanul." Gondolatmenetét elfogadva ugyanis hivatkozhatnánk egy közismert szoftverfejlesztő mamutcég már-már katasztrofálisan gyenge rejtjelezési megoldásaira is.

5. Visszatérek most a DES-ről és a Blowfish-ről tett kijelentésére. Idézem: „*... a komoly rendszerek (DES, Blowfish stb.) szinte támadhatatlanok a kódolt állományok oldaláról, viszont a szótáras visszafejtés oldaláról továbbra is támadhatók.*" Ez tévedés.

Mind a DES, mind a Blowfish az ún. „Feistel hálózatok" családjába tartozó rejtjelező algoritmus. Valamire is használható rejtjelező rendszerekben ezeket az algoritmusokat — a fentebb már említett szoftverfejlesztő cég egyes termékeit kivéve — kizárólag valódi (fizikai) és pszeudo (ál) véletlen egyszeri kulcsokkal alkalmazzák. (Minden rejtjelezés megkezdése előtt új kulcsot kell generálni. A kulcsismétlés gyakorlati valószínűsége majdnem nulla.) Ezért a „szótáras visszafejtés" (vagyis a *valószínű szavak módszerének* alkalmazása) értelmetlenné válik. A jól megtervezett Feistel hálózatokkal szemben hatékonyan felhasználható algoritmikus támadási módszer mai ismereteink szerint nem létezik. Egyetlen számba jöhető támadási eljárás a „brute force" (azaz *teljes kimerítés*). Szemben a DES-sel, a Blowfish sikeres megtámadása a brute force módszerrel — 128 bites kulcsméret mellett — ma még illúzió.

Az elmondottakból — remélem — világossá vált, hogy a javasolt rejtjelező eljárás sem szöveges, sem bináris fájlok védelmére nem alkalmas, még amatőr szinten sem. A rossz rejtjelező módszer használata sokkal nagyobb veszéllyel járhat, mintha nem rejtjeleznénk állományainkat. A rossz rejtjelezés ugyanis hamis biztonságérzetet ad, és ennek súlyos következményei lehetnek. „Rejtjelezz jól, vagy sehogy!" — mondták a régi korok kriptográfusai. Megszívlelendő tanács! A kulcs tehát valóban nem a lábtörlő alatt van, Ön azt a zárban felejtette.

Tisztelettel:

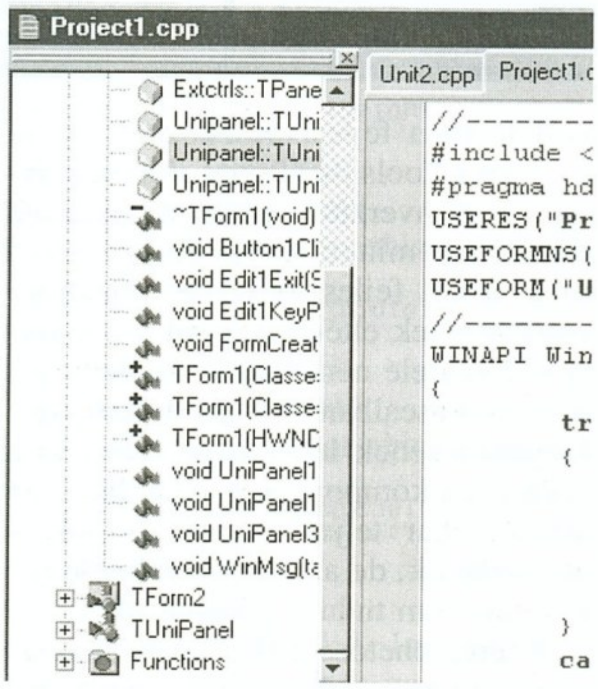
Kis Kálmán
kisk@montana.hu

Delphi 4 után C++ Builder 4

Látványos, magas C

A Windows programozásban annak idején korszakváltást jelentettek a vizuális fejlesztőeszközök. A 16 bites rendszerek programozásában mind a Microsoft által fejlesztett Visual Basic, mind a Borland Delphije sokaknak hozta életközelbe az üzenetvezérelt és objektumorientált programozás lehetőségét. Ugyanakkor a C/C++ nyelven fejlesztők csak a 32 bites világban kaptak az említett eszközökhöz hasonló fejlesztőeszközöket. Ezek egyikének, a Borland C++ Buildernek legújabb verzióját pedig a közelmúltban tették beszerezhetővé.

A korábbi verziók kiadásakor rendre az Object Pascal nyelvű Delphi fejlesztőkörnyezet C++ nyelvű megvalósítása történt meg. Ugyanez a helyzet a Borland C++ Builder 4.0 esetében is, amely megfelel a Delphi 4.0-s verziójának. Az új programkörnyezet jobb gépet igényel a fejlesztéshez, mint a korábbi C++ Builder. Az Inprise újjáélesztett Borland divíziójának (a továbbiakban egyszerűen csak Borland) ajánlata alapján a minimális igény egy 90 MHz-es Intel Pentium alapú számítógép, 32 bites Windows operációs rendszerrel. A telepítéshez javasolt szabad merevlemezterület legalább 175 MB, de a teljes telepítéshez közel 300 MB-ra lesz szükség. Ehhez célszerű hozzászámítani nemcsak a saját fejlesztések területigényét, hanem a Windowsok virtuális memóriahasználatának a helyigényét is (swap). Ez utóbbi jelentős lehet abban az esetben is, ha a javasoltnak megfelelő 32-64 MB RAM van a gépben. Tapasztalataim szerint egy egyszerű alkalmazás kialakítása és fordítása esetén a 64 MB-tal rendelkező, Windows 95-tel telepített gépen is 40 MB feletti swap terület lefoglalását igényli. Ha ez nem áll rendelkezésre, a látszólagos memóriahiány (hasonlóan a Delphi 4-hez) már a program indulásakor hibát okozhat. Az ilyen hibajelenségek egyik rendhagyó formája, amikor a fejlesztőkörnyezet nem, vagy csak részlegesen áll



fel, majd lebénul, és a memóriában hagyja a betöltött modulok egy részét, tartósan lefoglalva a gép erőforrásait. Ez azonban elkerülhető, ha a Windows virtuális memóriakezelését legalább 60-80 MB üres területtel rendelkező partícióra definiáljuk.

A program kényelmes használatához a Windows rendszerben természetesen szükségünk lesz egérre vagy más pixelvadász eszközre, és lehetőleg nagy felbontású monitorra. Már maga a rendszer telepítőprogramja is feltételezi a 600x800-as felbontást, mely a későbbiekben a különböző fejlesztőablakok áttekintésekor igencsak ajánlatos. (A CD-olvasó pedig ma már szinte kötelező.)

A C++ Builder 4 installálásakor a telepítőlemezről a C++ fejlesztőeszközön kívül közvetlenül a keretprogramból is telepíthető két másik program. Az egyik az InterBase 5.5, az Inprise által javasolt és lapunk márciusi számában ismertetett (kliens-szerver modellre alapozott) relációs adatbáziskezelő rendszer (Relational Database Management System — RDBMS), a másik a kész programok terjesztését szolgáló InstallShield Express telepítőkészítő. Már a program telepítésénél is találkozhatunk újdonságokkal a korábbi verziókhoz képest. Ilyen a Delphi 4.0-ban bevezetett és a Borlandnál korábban nem használt kulcskódos telepítés, illetve a frissítő verziók ellenőrző funkciója. A telepítőprogram ilyenkor bekéri a frissítési alapot jelentő korábbi verzió telepítő CD-jét, majd az ellenőrzést követően visszakéri az új verzióét, ami kétségtelenül kifinomultabb megoldás, mint elvárni az esetleg csak a letépett fóliaborításon meglévő, néhány év alatt majdnem bizonyosan elkallódó korábbi szériaszám begépelését.

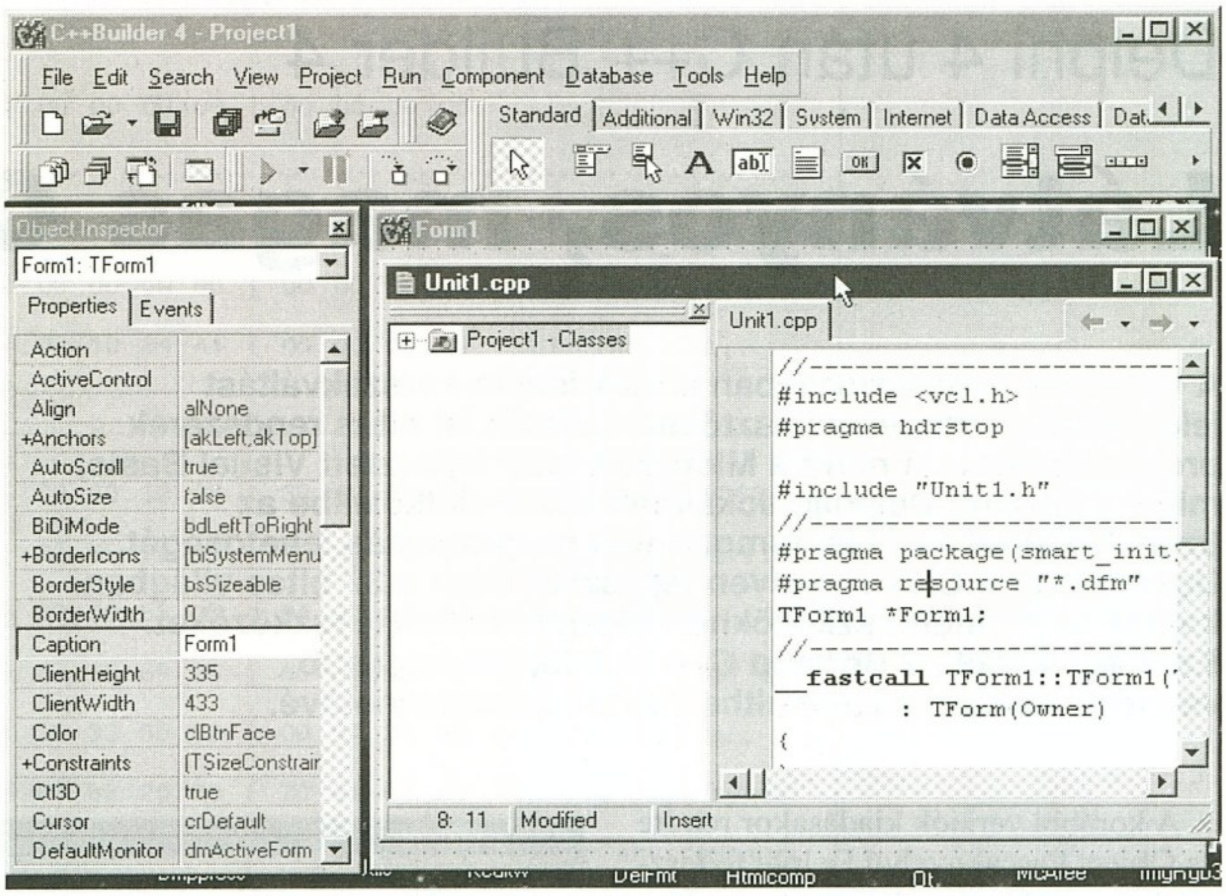
A program dobozában található CD-kollekcióban van a C++ Builder 4.0 próbaverziója is. Ahogy a felirat hirdeti: „give this to a friend", azaz másnak is odaadható kipróbálásra, bár ennek menete magyar viszonyok között kissé körülményes. A telepítő ugyanis először egy gépspecifikus azonosítót generál, majd ezt a Borlandnak elküldve megkapjuk azt a kódot (password), mellyel a 60 napig használható próbaverziót telepíteni tudjuk.

A végleges verzió telepítését követően jelenik meg a már említett törekvés, hogy az új C++ Builderrel a Borland deklaráltan a Delphi 4.0-ba beépített lehetőségeket kívánta elérhetővé tenni a C++ környezetben fejlesztők számára.

Ugyanakkor megmaradtak a korábbi verzió lehetőségei is. Ezek egyike, hogy személyre szabható a fejlesztőkörnyezet. A kódszerkesztő beállításaiban akár a DOS-os világban megszokott szövegszerkesztő tulajdonságai is beállíthatók, de mód van a funkciók gyors elérését lehetővé tevő eszközsorok egyéni kialakítására is.

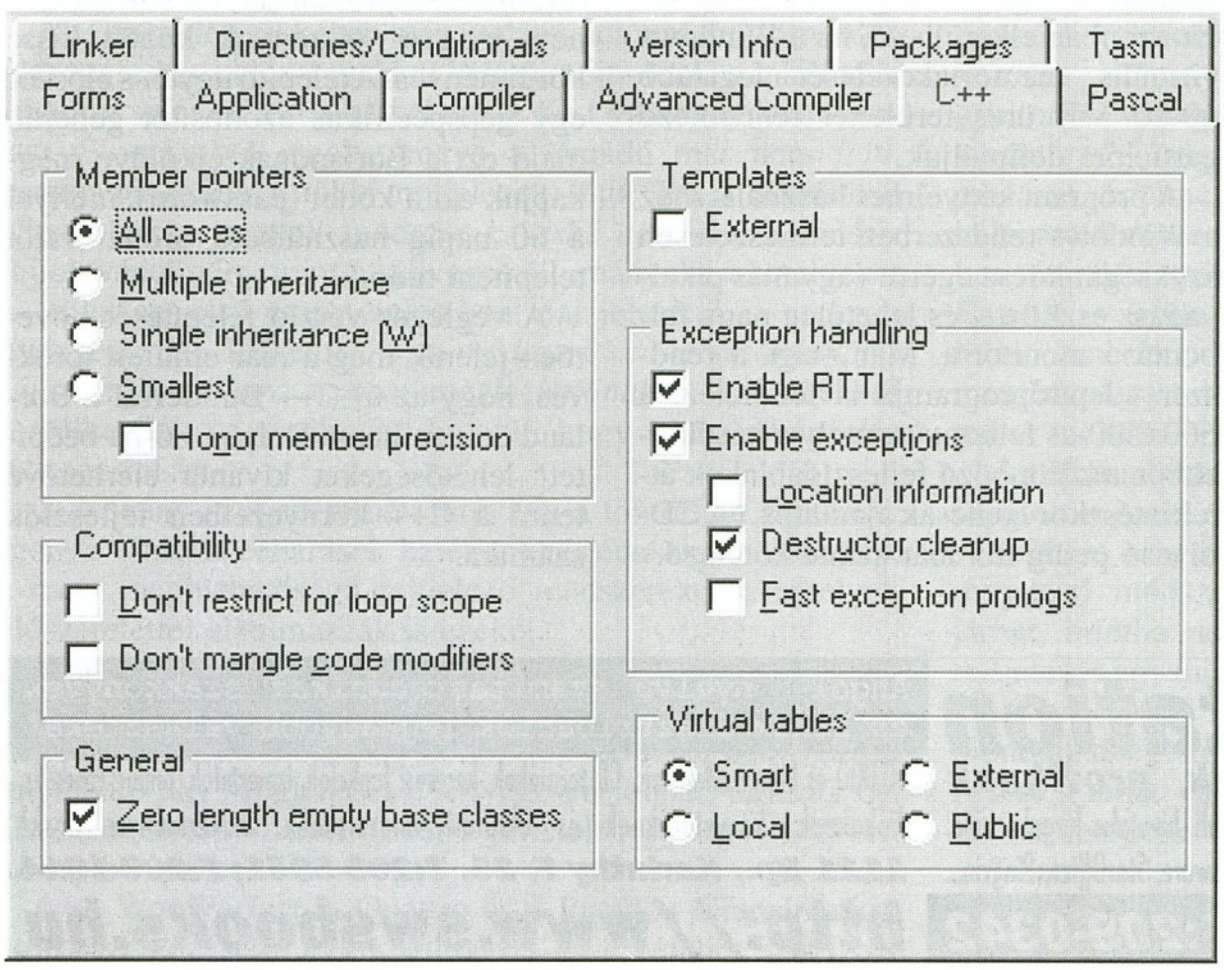
A C++ Builderre most áttérőknek továbbra is adott a lehetőség, hogy hasznosítsák delphis tapasztalataikat. A Delphiben fejlesztett komponensek átemelhetők a C++ környezet komponenspalettájára, amennyiben rendelkezünk azok forráskódjával. Ezeknek kompatibilisaknak kell lenniük a Delphi 4.0-val, amit az indokol, hogy abban, illetve a Builder 4.0-ban néhány objektum elérhetősége és használata megváltozott a 3.0-s verziókhoz képest. A változások elsősorban a Windows rendszerrel és más futó alkalmazásokkal intenzív kapcsolatokat teremtő programrészeket érintik. A Pascal források beépítésekor saját Delphi fordítót használ, így a Delphivel nem rendelkezők is felhasználhatják az ahhoz szép számban hozzáférhető forrásokat. A konverzió során a szokásos *.DCU fájlok mellett megjelennek tárgykódú fájlok (*.OBJ) és a használatukhoz szükséges fejlécfájlok (*.HPP) is.

A Delphiben programozókon kívül programjaikat most már azoknak sem kell újraírniuk, akik korábban a Microsoft Visual C fejlesztőeszközével dolgoztak. A C++ Builder külön segédeszközt biztosít az ebben készült projektek konverziójára, és ehhez ki sem



kell lépni a fejlesztőeszközből. Alapesetben a Tools menüpont alól meghívható a konverziót vezénylő varázsló. Sajnos az említett lehetőség egyirányú utca, a két fejlesztőeszköz objektumrendszerének eltérései miatt a konverzió visszafelé nem működik. De nincs meg a Pascalban elkezdett fejlesztés kiegészítésének lehetősége sem. Ha a szükséges komponensek rendelkezésre állnak, akár teljes Delphi formok is átemelhetők, de azokra további elemeket már nem tudunk ráhelyezni.

A fenti lehetőségnek a közös fejlesztés során mégis hasznát vehetjük, hiszen nem szükséges az egyes alkalmazásrészeket, ablakokat külön modulba,



például DLL-be befordítani ahhoz, hogy egy Builder projektben felhasználjuk. Ez pedig a fejlesztést, majd a kész alkalmazás telepítését jelentősen leegyszerűsíti. Kihasználhatjuk a Builder fordítójának képességeit is. (A tapasztalat szerint ugyanabból a forrásból kompaktabb kódot tudunk vele generálni, mint a Delphi fordítóprogramjával.) A fordítás paramétereit lehetőségünk van finoman hangolni a C++ és a Pascal forrásokra egyaránt. A komponenscsomagok használatával, pontosabban ezeknek a futtatható kódba való beépítésével a futtatható EXE fájl mérete néhány 10 KB-ra zsugorítható. A program a bináris kódú, DLL-ként viselkedő komponenscsomagokat külső egységekként kezeli.

A kódmegosztás nem érinti a hibakeresési lehetőségeket. A program 32 bites debuggerével (hasonlóan a korábbi verzióhoz) akár a CPU regisztereinek szintjéig megnézhető a program működése, és a DLL állományokban megkereshetők a hibák. A fejlesztett kódban feltételes töréspontok definiálhatók.

Ahogy a Windows platformot célzó fejlesztőeszközök többségével szemben elvárás, hogy támogassák a COM alapú alkalmazásfejlesztést, a C++ Builderrel is használhatók a COM szabványnak megfelelő külső egységek. Ezek az aktív vezérlők a Windows rendszerében vannak regisztrálva, ami jelentős könnyebbséget jelent külső komponensként való felhasználásukkor. Ugyanakkor ezeket a vezérlőket bármely programnyelven fejleszthetjük, és a komponenseket más forrásból is beszerezhetjük. Felhasználásukhoz nem is szükséges többet tudnunk, mint

a funkciók elvégzéséhez szükséges eljárások hívási konvencióit.

A C++ Builder nemcsak használja a Windowsban már regisztrált elemeket, hanem alkalmat kínál saját ActiveX komponenseink fejlesztéséhez is. Ezzel pedig ki-ki megjelenhet a komponenspiacon, mely az Internetnek köszönhetően egyre szélesedik. Az új aktív vezérlők készítésénél a Builder számára elérhető, a palettára is felvehető bármely komponenst át tudunk alakítani, akár a korábbi delphis fejlesztésekből származó egyedeket is. Lehetőség van teljes formokat ActiveX elemmé fordítani, melyeket később akár webdokumentumokba is beépíthetünk, feltéve, hogy a megcélzott böngésző támogatja ezek használatát. A kialakított komponenskönyvtárakat menet közben regisztrálhatjuk a Windowsba, így azokat kipróbálhatjuk más alkalmazásokból is. Ehhez az új komponensek könnyű kezelhetősége miatt felhasználható például a Visual Basic. (Nálam bevált.)

A Windows rendszer erőforrásainak használata azonban nemcsak egy már ott lévő komponens integrálását jelentheti, hanem új fejlesztőeszközökkel is kialakítható a dinamikus kapcsolat. Részben ezt jelenti a Windows üzenetek, API hívások kezelése, de lehetőség van más alkalmazásokkal dinamikus (például OLE kapcsolaton alapuló) adatcserét folytató alkalmazások fejlesztésére is, akár különböző gépeken zajló folyamatokat programozva. Ezt különösen a fejlesztőeszköz Enterprise változatával használhatjuk ki jól, a kliens-szerver architektúrájú programok fejlesztésekor.

Ha alkalmazásunk fejlesztése kinőné az egyedi fejlesztés lehetőségeit, a C++ Builder a csoportmunka támogatására is alkalmas. A program „magasabb" professzionális kategóriáiba beépített komplex verziófigyelés például azt jelenti, hogy projektcsoportok alakíthatók ki, melyekbe az egyedi projektek felvehetők a maguk alkalmazásaival és egyéb elemeivel. A fejlesztett alkalmazásnak a futtatását is megoszthatjuk, amire tipikus példa lehet az elosztott adatbázisokkal folyó munkát támogató alkalmazások készítése. Ezekben az esetekben előfordulhat, hogy az Interneten összekapcsolt gépek között is lehetőséget kell teremteni az alkalmazás és az adathozzáférés megosztására. Ehhez kapcsolati oldalon megvannak mind a hálózati eléréseket segítő, mind a HTML formátumot támogató komponensek. A hálózatos elemek pedig azonos programozási felületen teszik lehetővé a különböző hálózati megoldások (ISAPI, CGI, stb.) elérését, mert a komponensek az alkalmazáson belüli réteggel átfedik a különbségeket. Ugyanakkor a Midas és Corba alapú fejlesztések támogatásával heterogén hálózati környezetekben is boldogulhatunk.

A szabványosodó, sokszor a fejlesztőeszközök fejlesztői között is átfedést mutató lehetőségek kedveznek a programozói többnyelvűségnek, de felmerül a kérdés, hogy mit hoz a jövő a fejlesztőeszközökben. A Pascal és a C++ közös nevezőre hozásában a Borland mindenesetre nagyokat lép előre. Az Inprise Application Server magját képező AppCenter által megkövetelt Java-kompatibilitás kapcsán merült fel, hogy a Delphi és a C++ Builder fejlesztését is ezen kompatibilitás irányába kívánják terelni. A Java támogatása mellett talán a különböző nyelveken írt forrásokból bájtkódot generáló fordítók fogják megvalósítani a C nyelv eredeti célját, a hordozható kód előállítását a vizuális fejlesztőeszközök körében is.

Simay Endre István

Registry trükkök fejlesztőknek

Az Inprise-tól függetlenül is tovább finomíthatók a Borland fejlesztőeszközök beállításai. Marco Cantu, aki korábban könyvet publikált a Delphivel végzett programfejlesztésről, közzétett néhány Registry trükköt. Eredetileg a Delphi 4.0-hoz, de azok a C++ Builder 4.0-ban is működnek.

Az alábbi beállítások létrehozhatók a HKEY_CURRENT_USER\Software \Borland\Delphi, illetve HKEY_CURRENT_USER\Software\Borland \C++Builder bejegyzés alatt.

`\4.0\Extras\AutoPaletteSelect`

Típusa: String;
Értéke: „1"
Automatikusan váltja a komponenspaletta lapjait, hasonlóan egy menühöz.

`\4.0\Extras\AutoPaletteScroll`

Típusa: String;
Értéke: „1"
Az egérrel a komponenspaletta lapjainak végén található nyilak fölé állva automatikusan görgeti a komponenspaletta ikonjait, ha nem férnek ki.

`\4.0\Editor\DefaultHeight`

Típusa: DWORD;
Értéke: választható
Beállítja az Editor ablak nyitáskori magasságát.

`\4.0\Editor\DefaultWidth`

Típusa: DWORD;
Értéke: választható
Beállítja az Editor ablak nyitáskori szélességét.

`\4.0\Compiling\ShowCodeInsightErrors`

Típusa: String;
Értéke: „1"
Megmutatja a „Code Insight" hibákat a kódszerkesztő alatti üzenetablakban, amennyiben az nyitva van.

`\4.0\Component Templates\CCLibDir`

Megadható a hálózaton megosztott sablonok (Component Template) útvonala.

A Registry beállításainak átállításával egyszerűen megoldhatjuk például a kódszerkesztő színbeállításait. A tapasztalatok szerint, ha a HKEY_CURREN_TUSER\Software\Borland\Delphi\4.0\Editor\Highlight kulcs tartalmát exportáljuk egy szövegfájlba (*.REG), abban bármely szövegszerkesztővel lecserélhető a „Delphi" beírás „C++Builder"-re. Ha a változások elmentése után „visszaimportáljuk" a fájlt, közel azonos színbeállításokkal dolgozhatunk mindkét fejlesztőeszközben. Hasonló megoldással élhetünk más esetekben is, amikor a beállítások Registry kulcsai átfedik egymást.

(66) 77-90-95 / 30

A Fortran átváltozásai

A Fortran nyelv nem tartozik a „népszerű" nyelvek közé, abban az értelemben, hogy sokan és lezserül használnák. Ez a nyelv megmaradt a kutatóintézetek, egyetemek, tervezőirodák munkaeszközének. Persze, ha valaki éppen windowsos alkalmazást akar Fortranban programozni, azt is megteheti. A cikk felcíme nem telefonszám. 1966-ra, 1977-re, 1990-re és 1995-re utal, arra, hogy 30 év alatt ez volt az a négy kitüntetett év, amikor a Fortran nyelv egymást követő változatait szabványosították. Igen, szabványosították: a nyelvet ISO szabvány, sőt az USA-ban állami szabvány definiálja, nem csupán a megszokás.

Talán a szabványosságnak is szerepe van abban, hogy az igényesebb, főleg a sok számítással járó feladatok megoldásához a Fortran igen jó szerszám: az ember bízhat abban, hogy a sok ember-hónapi vagy ember-évi munkával kifejlesztett programot gépcsere után is használni tudja. Még akkor is, ha az új géppel új operációs rendszert kell üzembe helyeznie. Legfeljebb megválogatja, kitől veszi meg a Fortran fordítóprogramot. Néhány hónapja már a korábbi változatokhoz képest új filozófián alapuló Fortran 95-re is beszerezhető PC-s fordítóprogram, bár a PC-k felhasználói az új filozófiát nemigen veszik (vehetik) észre. (Lásd az Új Alaplap 1999. májusi száma Programozástechnika rovatában a HPF-ről szóló cikket.)

A fordítóprogram szót kezdetben szó szerint kellett érteni. Manapság a „compiler" sokkal több annál, bár a magyar terminológiában még mindig a „fordító" használatos. A ma megvett fordítóprogram testre szabott fejlesztőkörnyezet is lehet.

Fortran 66 — 60-as, 70-es évek

A ma rendelkezésre álló fejlesztési környezet értékeléséhez, megértéséhez vissza kell nyúlnunk a 60-as, 70-es évekből megmaradt emlékekhez. Ezek — már ahol még megvannak — irattárak polcain porosodnak. Szinte minden csak papíron érhető el, hiszen a felhasználói programok írói csak saját munkájukat őrizték meg, a rendszerprogramok a gyártók múzeumaiban találhatók meg. A 60-as évek második felében és a 70-es években Fortran 66-ot (régi nevén Fortran IV-et) is alkalmazó magyarországi számítóközpontok zöme csak közvetlen kollégáit szolgálta ki, idegeneket nem fogadtak. (Ez volt az „open shop", míg a „closed shop" üzemelésű számítóközpontokat bárki használhatta, akinek munkaszáma volt. Magyarázat: az „open shopba" beengedték a kollégákat, a „closed shopba" nem: ablakon át kellett beadni a munkafelvevőnek a programokat.) Így azután az Egyetemi Számítóközpont (ESZK) Razdan-3 gépe lett az első, amelyiken végre sokan megismerhették a Fortran 66-ot. (A gép 48 bites szószervezésű, 2 címes volt, 32 K ferritmemóriával, kb. 15-20 ezer utasítás/másodperc sebességgel. Inputja eredetileg lyukszalag és speciális lyukkártya, outputja sornyomtató. Háttértár: mágnesszalag.)

A Fortran fordítóprogramot az ESZK maga fejlesztette. Mivel a Fortran tipikus inputja az amerikai eredetű 80 oszlopos lyukkártya, saját erőből illesztettek a géphez ilyen kártyaolvasót. A kártyalyukasztást csak a „kiválasztottak" végezhették, akik a kártyalyukasztó használatát megtanulták.

Másoknak az 1. ábrán látható űrlapon kellett megírniuk programjukat, le kellett adniuk a munkafelvevőnél, majd másnap megkapták a sornyomtatott listát a hibákról. Ha a programozó elég gyors volt, még aznap leadhatta a javítást, és másnap újra mehetett a hibalistáért. Harmadik-ötödik fordulóra lett először szintaktikailag helyes a program, a szemantikai hibák keresése a második héten kezdődhetett. (A cikkhez tartozó ábrák helyhiány miatt a CD-mellékleten találhatók meg, FORKORNY.HTM néven.)

Az 1. ábrán az első programként írni szokott „Halló, világ!" látható, egy kis bonyolítással: a köszöntés a napszakhoz igazodik. A kötött formátumú Fortran programok a lyukkártya 7.–72. oszlopaira kerültek, az 1.–5. oszlopok fenn voltak tartva a címkéknek, a 6. oszlop az utasítások első sorában üresen maradt, míg többsoros utasításoknál ide bármit lehetett írni, rendszerint az 1, 2, 3 stb. sorszámokat. Hosszabb programok (néhány száz vagy pár ezer kártya) esetén a doboz kiborulása örökös rémképként lebegett a programozók szeme előtt — ezért azután a kártyalyukasztó lányok bosszúságára a 73.–80. oszlopokra be szokták írni a sorszámokat, például úgy, hogy a 73.–76. oszlopokon a program rövid neve álljon. (Így a „rendbetétel" a közönséges lyukkártyarendező gépekkel egyszerűen megoldhatóvá vált.)

A szintaktikusan már helyes programok első futtatásakor rendszerint kiderült, hogy a program hibás. Az 1. ábrán látható (szándékosan hibásan írt) mintaprogram célja az, hogy délelőtt „jó reggelt", délben „jó napot", délután „jó estét" kívánjon. Az (I1) input formátum csak a lyukkártya első oszlopát értelmezi, tehát a beolvasott szám mindig kisebb 10-nél, ezért a program mindig „jó reggelt" kíván. A javításhoz az ALAP9 jelű kártyát ki kell cserélni egy olyanra, amelyiken (I2) van. A módosításhoz ugyanazon a procedúrán kellett átesni, mind az előző fázisban. Nagy programok esetén a javításokkal hetek is elteltek.

Nem véletlen, hogy a Fortrant a praktikus amerikaiak úgy tervezték, hogy a programot részekre bontva lehessen lefordítani. Ha valamelyik szegmens már jó volt, kártyacsomagját félre lehetett tenni, míg a többi javítása folyt. Éppen ezért az ESZK később lehetővé tette, hogy az (európai eredetű) Algol programokat is lyukszalag helyett lyukkártyán lehessen fejleszteni. Akkor, amikor a Razdan-3 gépet kicserélték egy Robotron (NDK) R40 gépre, jó néhány, bárki által használ-

ható kártyalyukasztót is üzembe állítottak. Ezzel a helyzet csak látszólag javult, hiszen a gépkezelésben gyakorlatlan hallgatók tényleg sokkal több kártyát lyukasztottak, mint az adatrögzítő lányok, de a hibás lyukasztás aránya is megnőtt. Így azután az oktatói-kutatói feladatok esélyei romlottak.

Az ESZK ekkor már az Eötvös Loránd Tudományegyetem Számítóközpont rövidítése volt. A helyzet kilátástalanságát látva az ESZK pár év múlva üzembe helyezte az IBM TSO (Time Sharing Option) terminálrendszerét. Az R40-hez kötött 6-8 terminálon a felhasználók maguk szerkeszthették programjaikat és adataikat, s vagy interaktív módon futtattak, vagy gépre küldhették feladatukat kötegelt (batch) feldolgozásra. Utóbb néhány RJE (Remote Job Entry) terminált is felszereltek, amelyek otthon előkészített feladatok beadására szolgáltak. Ezek az RJE terminálok Commodore C-64 hobbigépek voltak — ki-ki munkahelyi, esetleg otthoni C-64-esén írta mágneslemezre feladatait.

Ez az időszak már átnyúlt a 80-as évek legelejére, amikor nálunk is megjelentek az IBM PC/AT gépek.

Fortran 77 — 80-as évek, 90-es évek eleje

Matematikai processzorral (Intel 80287) is felszerelt IBM PC/AT (80286) gépekre készült el 1984-ben az első PC-s Fortran fordító, a Ryan-McFarland-féle IBM Professional Fortran 77 compiler 1.00 változata (röviden PROFORT). A programok megírásához, javításához az IBM Professional Editor 1.00 változatát (röviden PREDIT) lehetett használni.

A PREDIT „egy az egyben" a TSO editorának PC-változata. A 2. ábrán látható a bejelentkező képernyője. (A program — vagy bármilyen más ASCII adatállomány — maga akármekkora lehet, mindig csak az éppen munkában lévő részét korlátozza a 64 KB.) A Fortran programokat a .FOR kiterjesztés jellemezte. A PREDIT-nek a kiterjesztéshez kapcsolódó, esetünkben FOR.PRF „profilja" voltaképpen a feladathoz összeállított makrógyűjtemény, például Alt-F beadása esetén a PREDIT a gyakran használt FORMAT szót szúrja be a szövegbe.

A 3. ábrán a már ismert mintaprogram látható a PREDIT munkaoldalán. A szövegből az első 72 oszlop látszik (a maximális sorhossz a sornyomtatókhoz igazodva 140). A Scroll-Lock bekapcsolásával a kurzormozgató nyilakkal lehet az ablakot a szöveg fölött mozgatni; egyébként az ablak automatikusan követi az írás helyét. A bal szélen látható kipontozott részre az F3 billentyűvel („LINE") lehet ugrani, ez a sorokra vonatkozó parancsok helye, például az A betűvel (after) jelölt sor után másolja be a C betűvel (copy) jelölt sort. Az F1-gyel lehet a funkcióbillentyűkkel elérhető menüket cserélni (például F1 egyszeri leütése után F3 „SAVE" jelentésűvé válik).

A részletek ismertetése nélkül is látszik, hogy ez az 1983-ban írt program valóban professzionális munkaeszköz. Sokszor ma is elő kell venni, hiszen például segítségével egy ASCII állományban — ilyenek a Fortran programok is — pillanatok alatt lehet a tabulátorokat (nem Fortran karakter) egyetlen T* LINE paranccsal kicserélni a megfelelő számú szóközre, hogy a bonyolultabb trükköket ne is említsük.

A cikkben hivatkozott ábrákat helyhiány miatt a CD-mellékleten helyeztük el, megtalálhatók a FORKORNY.HTM fájlban.

A PROFORT végső, valóban professzionális alakját később nyerte csak el, a Ryan-McFarland RMFORT 2.42 változattal, 1990-ben.

Az RMFORT 2.42 csomag része a FORTE program 1.01 változata (4. ábra), az *RM/FORTRAN Development Environment*. Ez már nevével is utal arra, hogy a Ryan-McFarland cég nem csupán a program lefordítását, hanem annak fejlesztését is gondozni kívánta. Mint az ábrán látható, a rendszer több „projekt" szimultán kezelését teszi lehetővé. Ezek alapértelmezésben mind a \RMFORT alkönyvtárban találhatók, de az ábra bizonysága szerint másutt is elhelyezhetők. A létrehozandó futtatható program neve projektnév.EXE lesz. Az 5. ábrán látható a projekthez tartozó Fortran programszegmensek (most csak egy) felsorolása, feltüntetve az utolsó módosítás és fordítás időpontját, a fordítóprogramnak az adott programhoz tartozó, nem alapértelmezés szerinti opcióit, a figyelmeztetések és hibák darabszámát, és azt, hogy az utolsó módosítást, ami még csak a FORTE munkanaplójában van előjegyezve, véglegesíteni kell-e.

A táblázat alatt jelennek meg a fordítóprogram üzenetei. A futtatási naplóból (6. ábra) kiolvasható a linker üzenetein kívül az is, hogy a program hibás: 17 órakor már nem „jó reggelt" kell kívánni. Ez a fentebb már említett ALAP9 sor módosítását igényli.

A szerkesztőprogram kezelése leginkább az IBM Personal Editorra (PE) emlékeztet (7. ábra). A módosított sorok vörös színnel, áthúzva megmaradnak a képernyőn, csak az editorból kilépve vesznek el. Undo lehetőség természetesen van. (Az ALAP9 kártya módosítása könnyen azonosítható az ábrán. A kártyasorszámokra a PC-ken már nincs is szükség, ezért a képernyőn alapértelmezésként csak az 1.–72. oszlopok láthatók.) Az esetleges diagnosztikai üzenetek (pl. két egymás melletti „+" jel szintaktikus hiba) a szerkesztésre kiválasztott szegmensbe, a megfelelő helyre beírva is megjelennek.

A Lahey Computer Systems Inc. Fortran 77 fordítóprogramjának (F77L-EM/32 Ver. 4.01) fejlesztőrendszere előfutára a Lahey Fortran 90-esének, ezért részletes ismertetésére nem térünk ki. Egyetlen jellemző tulajdonságát viszont érdemes kiemelni: a 8. ábrán tanulmányozható a DOS alatt futó Lahey Blackboard Editor néhány, annak idején a programeditoroknál egyáltalán nem tipikus szolgáltatása, DOS alatt futó ablakos (window) rendszerben.

Végül néhány szó az IBM Fortran 77 fordítójának legkönnyebben elérhető változatáról. Ez az RS/6000 munkaállomásokon fut, AIX operációs rendszer alatt. Ez a nagy teljesítményű operációs rendszer igazán multiprogramozású, de sajnos Unix-változat. Ez a tervezési filozófia azzal jár, hogy a programfejlesztésnek jóformán nincs is más eszköze, mint a (szerintem megjegyezhetetlen) rövidítésekkel vezérelhető „vi" editor.

És a Microsoft?

A Ryan-McFarland fejlesztői környezettel nagyjából egyidős a Microsoft rendszere (1991). A nagyobb cégtől több, még inkább felhasználóbarát szolgáltatást várhatnánk, mint a kisvállalkozásként működő RM-től. Amint azt megszokhattuk, a Fortran esetében is hiába várunk.

Az IAEA (az ENSZ szakosított szerve) megbízásából, nemzetközi terjesztésre fejlesztettünk egy Fortran programcsomagot. A csomag legnagyobb programja kereken hatezer soros (amibe nem számoltuk bele a több program által használt, külön lefordított szubrutinok hosszát). Ezt az

RMFORT 2.42 könnyedén lefordította. Mivel a jövendő felhasználókról nem tudhattuk biztosan, hogy megvan-e nekik az RMFORT 2.42, más fordítóprogramokat is használnunk kellett. A Lahey Fortran 77 fordítóval szintén gond nélkül ment minden. Ezzel szemben a Microsoft FL programja ötödik (5.00) változatának a program túl nagy falat volt. A következő üzeneteket kaptuk:

```
„program too large"

„unrecoverable heap overflow"

„local symbol table overflow"
```

Fel kellett a programot darabolni kb. ezersoros részekre, hogy a Microsoft FL 5.00 egyáltalán kezelni tudja. (A legfrissebb, 5.10 változat sem hozott javulást.)

A Microsoft Fortran fejlesztőrendszerének keretprogramja a CodeView. Ennek képernyőformátuma a 9. ábrán látható. Feltűnő, hogy a kép alján hexadecimális értékek vannak felsorolva: ezek a soron következő gépi utasítások — még csak nem is Assembly nyelven. A kép jobb szélén a processzor regisztereinek tartalmát láthatjuk. Már ennyi is elég annak igazolására, hogy a CV egyáltalán nem volt kényelmesebben kezelhető, mint a DOS DEBUG programja: a Fortran nyelven programozóknak is a gépi kód szintjéig kellett elmerülniük a hibakeresésben.

A processzor regisztereinek a programozó általi ellenőrzése már csak azért is fontos volt, mert a Fortran fejlesztőrendszerben egy igen súlyos hiba volt: a linker a szegmensregisztereket hibásan kezelte. A memóriacím a szegmensregiszter tartalma tizenhatszorosának (10H) és a címregiszter tartalmának összege. A linker akkor, ha a „misztikus" /Gt5 kapcsolót nem használtuk, független adatmezőket címzett ugyanarra a helyre, aminek következményeit megjósolni sem lehetett. A bogarászás során gyakran ilyen eredetű programhibákat kellett megtalálni. „Misztikus" kapcsoló, mert a dokumentációból nem derült ki, miért éppen 5-ös számot kellett ide írni. (Részletesebben lásd: Alaplap, 1992. október, 54. oldal.)

És ez még nem minden. Vélhetően üzletpolitikai megfontolásból az FL 5.00 és 5.10 nem kompatibilis a két másik említett Fortran-fordítóval, de a Fortran-szabvánnyal sem. Ha valamit nem tud, az még megbocsátható. Ha többletszolgáltatásai vannak a szabványhoz képest, legfeljebb nem használjuk azokat, hogy a programot más környezetben is le lehessen fordítani. Az viszont elfogadhatatlan, hogy szabványban definiált kifejezést más jelentéssel használjanak. Emiatt a Microsoft Fortran-fordítóját legfeljebb olyan apró számításokra használjuk, amit akár a DOS részeként kapott QBasickel is meg lehetne oldani. (Az már csak apró kis tüske, hogy karakterkészlete is eltér a szabványtól; a Fortranban nincs, a Microsoftnál van szögletes zárójel, [ ], és nem is lehet kikerülni a használatát.)

És a fenti felsorolás még messze van a teljességtől. Félő, hogy nemcsak a mi programunk lefordítását, hanem a nemzetközileg elfogadott tesztfeladatokat sem lehet az FL 5.00 (ill. 5.10) fordítóprogram használatával megoldani. A következmény: a Microsoft Fortran fordítóprogramja nem szerepel a Weben elérhető, a fordítóprogramokat fejlesztő cégektől független Polyhedron Software Ltd. 1998. évi benchmark teszt eredménylistáján (http://www.polyhedron.com).

Fortran 90 — a 90-es évek közepétől

Az University of Linköping, National Supercomputer Centre (http://www.nsc.lu.se) által összeállított listában és a Weben az „F90 FAN's: Fortran 90 Frequently Asked about News" címmel (ftp://ftp.ifremer.fr/ifremer/ditigo/fortran90) alig néhány Fortran 90 fordítóprogram található.

Ezek ábécérendben:

Absoft (http://www.absoft.com) — Power Macintosh; Unix, Linux.

Apogee Software Inc. (http://www.apogee.com) — Sparc.

Cray (http://www.cray.com/products/software/pe) — Cray YMP, Superserver 6400, Sparc Solaris 2.3.

Digital Fortran, új neve Compaq Fortran (http://www.digital.com/info/hpc/f90) — Digital Unix Alpha és OpenVMS Alpha.

Fujitsu (http://www.fujitsu.co.jp/hypertext/InfoPro/opsp/supports.html) — Sparc Solaris, Sun Sparc, HP PA-RISC.

IBM (http://www.software.ibm.com/ad/fortran/xlfortran) — XLF (XLHPF), RS/6000, AIX.

Imagine1 (http://www.imagine1.com/imagine1) — F (ingyenes Fortran 90 kivonat), Linux.

Lahey Computer Systems Inc. (http://www.lahey.com) — LF9040 (Fortran 90), PC, DOS; LF9505 (Fortran 95), PC, Windows 9x; ELF90 (Fortran 90; a fejlesztőrendszer nélküli változat ingyen letölthető; részletes ismertetés: Új Alaplap, 1998. augusztus, 66. oldal); mindhárom rendszer PC-n fut.

NA Software (http://www.nasoftware.co.uk) — PC, Linux; Sun Sparc, IBM SP2.

NAGWare (http://www.nag.co.uk/nagware) — Sparc Solaris; Linux.

Pacific Sierra (http://www.psrv.com) — Linux; a Unix, VMS, Convex változat Fortran 77-re fordít.

The Portland Group (http://www.pgroup.com) — SGI, IBM SP2, HP/Convex, Linux.

Salford (http://www.salford.ac.uk/ssl/ss.html) — FTN 90: a NAG fordító PC-re átdolgozva; FTN95: DOS, Windows 3.1, Win32 (NT, 95).

SGI (http://www.sgi.com) — R8000 (Power Challenge, Power Indigo 2, Power Onyx) Irix 6.1 alatt; Cray.

Sun (http://www.sun.com) — Sparc, Cray.

Látszik, hogy a Fortran 90 a munkaállomások és a szuperszámítógépek nyelve. A PC-n is futtatható Absoft, Imagine1, NA Software, NAGWare, Pacific Sierra és Portland Group fordítók Linux környezetet igényelnek. A Salford fordítója eredetileg Sparc Solaris munkaállomásra készült.

Ennek ismeretében még inkább nagyra értékelhető, hogy a Lahey rendszerét eredetileg is PC-re dolgozták ki (a processzor lehetőségeinek teljes kihasználását azzal érték el, hogy a munkában az Intel is részt vett). A hatékony programozás érdekében a Soft As It Gets ausztráliai cég (http://www.getsoft.com) ED editorát hozzáigazították a Fortran követelményeihez, beleértve a windowsos programok írásának nem tipikus Fortran igényét (WiSK, lásd alább). Ez a Lahey ED4W „editor" (10. ábra) mindent felkínál, mi „szem s szájnak ingere" (11-12. ábra). Az ED4W fejlesztőrendszer próbaverziója letölthető a Lahey http://www.lahey.com honlapjáról. — Nem lehet megállni megjegyzés nélkül: egy szakértő cég nem akar mindent maga megcsinálni, elismeri, ha valamit más jobban tud, hiszen saját területén a legjobb. Nos, az FL+CodeView programcsomag teljes egészében a Microsoft terméke. Egyik komponense sem teljesértékű.

Az ED4W-t használva elegendő a feladathoz tartozó programot tartalmazó alkönyvtárat kiválasztani, utána az AUTOMAKE gombra kattintva a Polyhedrontól származó AUTOMAKE „összekeresi" a szükséges szegmenseket, majd a fordítóprogram lefordítja őket, és a linker összeállítja a futtatható programot. Az << ill. >> gombokkal lehet végig-

futni a fordítóprogram üzenetein: két ablakban (11. ábra) egyszerre láthatjuk az üzenetet és a forrásnyelvű lista megfelelő sorait. (A példában az üzenet a Fortran 66 stílusú programozás elavultságára figyelmeztet, de a „backward compatibility" miatt csak külön kívánságra tekinti a fordítóprogram az elavult szerkezeteket egyúttal hibásaknak is.) Az üzenetben említett sort széles vörös sáv teszi könnyen megtalálhatóvá. Egyébként a színkódok széleskörűen használhatók, a felhasználó által ízlés szerint módosítható színek emelik ki a Fortran alapszavakat, a kommentárokat stb. A jobb egérgombbal egy szerkezet nyitóelemére kattintva ráugrik a záróelemre (például bal és jobb zárójel, DO-END DO, INTERFACE-END INTERFACE stb.), tehát könnyű megtalálni az esetleges elírásokat.

A DOS alatti interaktív programfuttatás programozását segíti az ED4W-nek az Interactive Software Services INTERACTER szubrutincsomagját (http://www.demon.co.uk/issltd/) kezelő LiSK (Lahey Interacter Starter Kit) modulja. A fejlesztőrendszerből elérhető a WiSK (Windows Interacter Starter Kit) szerkesztőprogramja, amivel Fortranban lehet meglepően gyors (a szokásosnál legalább kétszer gyorsabb) Windows alkalmazásokat programozni. A „Halló, világ!" programot csak erőltetetten lehetne WiSK példaként bemutatni, ezért a 13-14. ábrákon a Lahey demói láthatók. Mindkét modul része a megfelelő grafikai szubrutincsomag is. (Ezt használva a CalComp stílusú régi rajzolóprogramokat is lehet kezelni.)

Az ED4W természetesen futásidőben is hatásos: a DEBUG gombbal elindítva a SOLD programot (Lahey Symbolic On-Line Debugger), az a felhasználó által írt program futását úgy szervezi, hogy mindazt elvégezhetjük, amit assembler programokkal tehetünk a DOS DEBUG parancsát használva. Ráadásként egy külön ablakban az éppen aktív programrész forrásnyelven követhető (15. ábra).

És még mindig nincs vége. A Lahey Fortran lehetővé teszi a kevert nyelvű programozást (például Assemblerben, C-ben, vagy éppen Pascalban írt szubrutinok hívását). Ezt az ED4W is támogatja. Az itt következő hosszú lista csak azoknak a nyelveknek a felsorolása, amelyek szabályait, szokásait a kiterjesztés alapján (12. ábra: „kiterjesztés konf.") az ED4W érvényesíti. Van köztük olyan is, amelynek nálunk még a neve is ismeretlen.

Ada, Assembler, Basic, C/C++, Clipper 5, Clipper S87, Cobol, DataFlex, dBase, DOS Batch, Eiffel, Forth, Fortran, FoxPro, HTML, Ini (Windows), Java, Lisp, Modula 2, „None" (nincs szabály), Paradox, Pascal, Perl, PL1, Progress, Prolog, Rexx, Smalltalk, SQL, Text, Verilog, VHDL.

A munkát egyszerűsítő „napi tippet", amivel az ED4W bejelentkezik, le lehet ugyan tiltani, de akkor a jó kedélyhez adandó jó tippeket is elveszítjük, például ezt is: „There are 3 kinds of people: those who can count and those who can't."

Fortran 95 — 1998 vége óta

1998 novemberében jelentette be a Lahey cég a Fujitsuval közösen fejlesztett Fortran 95 fordítóprogramjának kibocsátását. Sajnos a kincstári beszerzési előírások miatt e cikk nyomdába adásáig (április vége) sem sikerült a 90-95 upgrade beszerzése, ezért az alábbiakban írottak mögött nincs saját tapasztalat.

Jóllehet a Fortran 95 kevés nyelvi újdonságot tartalmaz a Fortran 90-hez képest, ennek a fordítóprogramnak a jelentősége rendkívül nagy: az általánosan elterjedt Fortran fordítóprogramok kb. 6 és 13 decimális jegy (szimpla és dupla) pontosságú lebegőpontos számítást tesznek lehetővé. Az új fordítóprogram ismeri a 33 számjegyes (négyszeres) pontosságot is, az ábrázolható számok tartománya pedig $\pm 10^{\pm 4900}$. A szokásos 8, 16 és 32 bites egészeken kívül 64 bitesek is vannak (±9,223,372,036,854,775,807). A Fortran 90 fordítóval is kezelhető programok futásideje a 95-össel fordítva kb. felére csökken!

A fejlesztőkörnyezet „standard" változata ugyanazokat a programokat tartalmazza, mint a Lahey Fortran 90-é. A „pro" változat ezeken kívül magába foglalja a Fujitsu Visual Analyzert (16. ábra), a Fujitsu C nyelvi rendszert és a Fujitsu SSL2 matematikai könyvtárat. A könyvtárban több mint 250 szubrutin található a lineáris algebrai feladatok megoldására, sajátértékek és sajátvektorok számítására, nemlineáris egyenletek megoldására, szélsőértékek keresésére, függvények interpolálására, Fourier- stb. transzformációkra, numerikus differenciálásra és integrálásra, differenciálegyenletek megoldására, speciális függvények számítására és véletlen számok generálására.

Azért, hogy a korábban már megoldott feladatok ismételten felhasználhatók legyenek, a rendszer támogatja a Fujitsu C, Microsoft Visual C++ (≥ 2.0) és a Borland C++ (≥ 4.5) nyelvű lefordított alprogramok hívását. Microsoft Visual Basic, Microsoft Visual C++, Borland C++, és Borland Delphi alapú DLL-ek hívhatók a Fortran programból. Hasonlóan, közvetlenül hívhatók a Win32-es API-k.

Végül az LF95 telepítésének hardveres és szoftveres feltételei:

CD-ROM meghajtó; 80486DX, Pentium, Pentium Pro, Pentium II vagy kompatibilis processzor; 24 MB RAM; 50, ill. 60 MB HDD terület a standard, ill. pro változathoz: Windows 9x vagy NT.

Régi programok konverziója

A Fortran 90 szabvány egyes Fortran 66 megoldásokat elavultnak minősített. Ezek a Fortran 95-ben már hibának számítanak. Másrészt egyes Fortran 77 elemeket a Fortran 95 ítél elavultnak. Ha ki akarjuk használni a Fortran 95 előnyeit, a korábbi programokat esetleg jelentősen át kell dolgozni.

A már említett AUTOMAKE kidolgozója, a Polyhedron kínálja erre a feladatra a SPAG programot, ami a régi „spagetti"-bonyolultságú programokból néhány Fortran 90 elemet is tartalmazó, strukturált Fortran 77 programot hoz létre. Néhány érdekesebb beavatkozás:

— opcionálisan törli a végrehajtásra soha nem kerülő programrészeket,

— törli a deklarált, de nem használt változókat,

— használhat kis- és nagybetűs írásmódot az egyes változótípusok megkülönböztetésére.

Két példát mutat be a SPAG munkájára a CD-melléklet SPAGETTI.HTM állománya. A SPAG-ot egészíti ki a plusFORT minőségbiztosítási rendszer. Ennek keretében az analizálandó program minden részlete olyan statikus vizsgálat alá kerül, mintha végre lenne hajtva, így például ellenőrizhető az alprogramok formális és aktuális paramétereinek megfelelősége.

Az ezt követő dinamikus vizsgálat keretében a plusFORT elkészíti a program tesztváltozatát, ami valódi adatokkal futva naplózza a futás folyamatát, például jelezve a nem definiált értékű változók használatát. Kiderül, hogy a program mely részei futnak ritkán, melyek gyakran — az utóbbiak optimalizálására kell rászánni a többletmunkát.

Ahogy a Fortran fejlesztőkörnyezet fejlődik, úgy lesz egyre kevesebb manuális (többlet)munkára szükség.

Szondi Egon János
szondi@reak.bme.hu

E SZÁMUNK HIRDETŐI

Cég	Info#	Old.
2F	01	24.
Allegro	02	23.
Areco Informatika	03	35.
Borland	04	06.
Business Online	05	72.
Computer Panoráma	06	71.
ComputerBooks	07	71.
Corg Computer	08	35.
DIT Digitáltechnika	09	35.
Elender	10	B2.
Hewlett-Packard	11	B4.
KerSoft	12	06.
Keszo	13	73.
Next	15	06.
Onyx	16	06.
Onyx	17	71.
Oracle	18	04.
Profon	19	35.
Qwerty	20	58.
Ready	21	35.
Server	22	24.
Software Station	23	61.
StarOffice	24	51.
Telnet	25	42.
VirusBuster Team	26	24.
VTCD	27	B3.

A Mikrobazár rovatban a nem kereskedelmi célú egyéni hirdetések közlése ingyenes.

A kereskedelmi célú apróhirdetések tarifája gépelt soronként (azaz 60 karakterenként) 300 forint.

A terjedelem alapján így kiszámított összeget kérjük átutalni az Új Alaplap Kiadói Kft számlájára (OTP, 11706016–20788599), vagy feladni postai utalványon a kiadó címére (1539 Budapest, Pf. 571), és feltüntetni, hogy „Új Alaplap, apróhirdetés". A befizetést igazoló szelvény másolatát — a hirdetési szöveggel együtt — a szerkesztőséghez (a kiadóéval azonos címre) küldjék el.

Szerzői jogokat sértő szoftverhirdetéseket nem közlünk le.

Bármilyen típusú szöveg fordítását vállalom angolról magyarra, magyarról angol nyelvre, illetve vállalom kiadványok látványtervezését, szerkesztését is. Cím: Lachner Zoltán, 1195 Budapest XIX., Jáhn Ferenc u. 14/a. Telefon: 357-0308.

OBJECTS 2.0 — objektumorientált programozás CLIPPER-ben. Tájékoztató kérhető az alábbi címen: Szűcs János, 4400 Nyíregyháza, Vasvári Pál u. 37. Tel.: (42) 437-331 vagy 465-666/1382-es m.

Adatmentés CD-re, streamerre; winchesterről, floppyról. Ugyanitt beszerzési tanácsadást, hálózattervezést és programkészítést is vállalok. Cím: Kovács Lajos, 1031 Budapest III., Vízimolnár u. 10. IV/33.

Alaplapcsere, memória-, winchester- és floppybővítés a helyszínen. MegaSoft. Telefon: 295-5085.

Stúdiómban megbízhatóan, ellenőrzötten lefordítom angol, német, francia és magyar nyelvről/nyelvre műszaki és közgazdasági folyóiratok cikkeit, hardver- és szoftverleírásait. Áfás számlát állítok ki. Cím: Szász György, 1035 Budapest III., Kórház u. 25. Tel.: 368-4874.

Akarod, hogy ingyen tiéd legyen az évszázad viccgyűjteménye? Nos, ha igen, akkor írj a kgb11@egon.gyaloglo.hu címre, vagy keress telefonon (20)970-3438!

Keresem az Apogee cég által kiadott Paganitzu /Paga1/ játék második és harmadik részét, melynek címei: II. Quest for the silver dagger és III. Jewel of the Yucatan. Telefon: 250-2150

Elcserélném vagy jelképes összegért átadnám magyar nyelvű Commodore 64-es irodalmaimat. Cím: Kovács Gábor, kgb11@egon.gyaloglo.hu. Telefon: 06-20-970-3438.

Keresem az "IDEAL" papírvágó gépek szervizháttérrel is rendelkező magyarországi forgalmazóját. Aki tájékoztatást tud adni, kérem, hívja a 339-8515-ös telefonszámot!

Compaq Armada típusú P120-as laptop (1 GB HDD, 16 MB RAM) eladó. Irányár: 180 ezer Ft. Érdeklődni: Takács Péter, (22)313-052 (napközben).

Eladó gép! Nagy toronyházban 133-as Pentium, 32 MB EDO RAM, 4 MB Virge VGA kártya, 120 MB és 2,5 GB Winchester (az utóbbi még garanciális), Sound Blaster Pro hangkártya, 14"-os SVGA Cordata monitor, 3,5"-es és 5,25"-es floppymeghajtó. 55 000 forint. Telefon: 357-0308.

Programfejlesztő szakembereket keresünk **legális ausztriai (bécsi) munkára, későbbi otthoni folytatással.** Feltételek: német nyelvtudás, minimum 1 év gyakorlat, általános programozói ismeretek. Lehetőségek: nagyon magas kereset, nyelvtudás elmélyítése, külföldi tapasztalatszerzés, látókör bővítése. Jelentkezés e-mailben (info@psi.co.at), telefonon (00/431/9117757) vagy levélben (Austroex-Bohn, A-1140 Wien, Hütteldorfer Strasse 193/24).

Permanens szakmai nyelvújítás

Hétfejű terminológiai sárkány

Állandó változásban, fejlődésben lévő szakterületünknek a kezdetektől fogva neuralgikus pontja a korszerű szakmai szókincs kialakítása, és persze szótárakban, számítógépes rendszerekben való megjelenítése. Világprobléma a szakmai terminológia viszonylagos kialakulatlansága, és menet közben vagyunk kénytelenek jól-rosszul megoldani saját problémáinkat. Ha állandó késésben vagyunk is, nem szabad nagyon lemaradni. Az állandó pezsgést, megujulást tükröző angol terminológia többé-kevésbé adekvát anyanyelvi tükrözése kényszerítő szükségessége előrehaladásunknak.

Mik is a szakmai terminológia problémái? Próbáljuk meg összeszedni a nehézségek legfőbb okait!

1. Egyre újabb fogalmak, és velük együtt újabb szavak, kifejezések jelennek meg a szakmai irodalomban.

2. Idővel a meglévő szavak, kifejezések jelentéstartalma is bővül, szűkül, vagy módosul, átalakul.

3. Gyakran sehol nem lehet megtalálni a szavak, kifejezések pontos meghatározását, ehelyett a szövegösszefüggésből lehet (jól vagy rosszul) kikövetkeztetni.

4. A különböző szerzők által használt kifejezések jelentéstartalma között sokszor felfedezhető a részleges átfedés, de a pontos értelmezést nem kapjuk meg. Gyakran még ugyanaz a szerző sem használja mindig azonos értelemben ugyanazokat a kifejezéseket.

5. A különböző gyártó cégek között, sőt egyes cégeken belül sincs egység és következetesség a szóhasználatban. Arra is van példa, hogy egyazon „hivatalosnak" tekinthető dokumentumon belül következetlen a terminológia. (Például azért, mert más személyek írják az egyes részeket, és elmulasztják az egyeztetést.)

6. Hasonló jelenség nem ritka a magyar fordításban megjelenő művek szakmai nyelvezetében sem. Találkoztam olyan esettel, hogy két kiváló szakember teljesen eltérő szakmai terminológiát használt egyazon könyvön belül, feltehetően ugyanazokra a jelenségekre. Ha közben a fordítók nem adják meg zárójelben az eredeti angol elnevezéseket, akkor az olvasó teljes bizonytalanságban marad, hogy valóban azonos fogalmakról van-e szó.

7. Rengeteg rövidítést használnak a szakmában. Ezeknek gyakran még a kifejtése sem egyszerű. Előfordul, hogy a számítástechnika egyik részterületén teljesen más értelemben használják ugyanazt a rövidítést, mint a másikon.

8. A hardveresek és szoftveresek szóhasználatában is előfordulnak — néha nem is lényegtelen — eltérések.

Némi töprengenivaló

További három tényező megérdemli, hogy bővebben is kitérjünk rájuk, mivel megértésük komoly átgondolást igényel.

a) Gyakran ugyanannak az angol terminusnak többféle fordítása létezik, ki ezt használja, ki azt. Előfordul, hogy megérteni is csak akkor sikerül egyik-másik szerző mondanivalóját, ha ki tudjuk deríteni, milyen angol szó fordításaként használta ezt vagy azt a kifejezést.

Egy-egy terminussá váló szó vagy kifejezés keletkezése néha véletlenszerű. Előfordul például, hogy valamelyik fordító spontán ötletéből fakad, mint valamilyen terminusnak az adott szövegösszefüggésbe jól beleillő fordítása. Ebből azonban még nem következik, hogy a találó elnevezés az adott idegen nyelvű terminus univerzális helyettesítésére is alkalmas, nem biztos, hogy más helyzetekben is jól kifejezi annak jelentéstartalmát. De a fordítottja is megesik, az, hogy egész iskolák hadakoznak egy-egy újonnan gyártott anyanyelvi terminus befogadásáért vagy befogadása ellen. Elég, ha csak az ősidőkben lefolytatott számológép/számítógép vitára vagy az újabb kori file/fájl/állomány/telep stb. körül kibontakozott pengeváltásokra utalunk.

De nem is az az érdekes, hogy véletlenül vagy tudatos aktus következtében született-e valamelyik terminus, e, és hogy mennyire szerencsés vagy nem szerencsés a példaként tekintett terminus hű visszaadására. Sokkal fontosabb kérdés, hogy valamilyen (lehetőleg egységes) megállapodásra jusson a szakma, és hogy az olvasók megértsék. Ehhez viszont az kell, hogy jelentéstartalma elég pontosan körül legyen írva, lehetőleg példákkal illusztrálva, és hogy ez a leírás hozzáférhető legyen bárki számára.

b) A szakmai terminológia forrása az angol nyelv. Elég, ha egyetlen következményére gondolunk ennek a helyzetnek. Az angol szakmai szókincs kialakítói — bármelyik szakterületről legyen is szó — szívesen merítenek a köznyelvből. Ennek viszont egyik legfontosabb jellemzője, hogy eleve rengeteg benne a többértelmű szó, kifejezés. A „köznyelvi áthallás" egyáltalán nem zavarja az angolokat, sőt azt tartják, bizonyos mértékig segíti is a megértést.

De van ezzel homlokegyenest ellenkező felfogás is. A német típusú gondolkodás szerint például a nyelvújításnak ez a módja komoly veszélyforrást jelent, hiszen mindenki azt értheti bele az olvasott vagy hallott kifejezésbe, amit akar, vagy amit egyéni ismeretei determinálnak. Szerintük tehát lehetőleg kerülni kell a köznyelvi szavak átvitelét a szakmai nyelvbe, mert ez félreértésekhez és félremagyarázások-



hoz vezethet. E felfogás szerint nem jó az, ha a konyhanyelv lezsersége átszüremkedik a szakmai szóhasználatba, mert megzavarja a terminus értelmének pontos körülhatárolását.

Felmerül tehát a kérdés: melyik felfogást kövessük? Átvegyük-e az angol nyelvben uralkodó könnyed és játékos nyelvújítási szellemet, vagy a német precízségre törekvést tekintsük példaképünknek.

c) A „követő nyelvújítás" talán még nehezebb feladat, mint egyetlen nyelven belül megküzdeni a terminológiai kihívásokkal. Magunkra konkretizálva a tennivalókat: az angol nyelv vonatkozásában nekünk nemcsak az a feladatunk, hogy meghatározott fogalmak kifejezésére alkalmassá tegyük saját nyelvünket, hanem például az is, hogy a tükrözés lehetőségeinek megteremtése céljából bizonyos mértékig illeszkedjünk a másik nyelv adottságaihoz, szerkezeti jellegzetességeihez.

Egy példa talán világosabbá teszi, mire gondolok. Tudjuk, hogy az angol nyelv szinte korlátlanul használja ugyanazt a változatlan alakot igének, főnévnek, gyakran még melléknévnek is. Nekünk ez nehezebben megy, általában a szótő továbbképzésével vagy szóösszetételekkel tesszük eltérővé a nyelvi alakot. A megfelelő szótő kiválasztásakor tehát arra is gondolnunk kell, hogy biztosítva legyenek a magyar megfelelőként kiválasztott szó továbbképzési, összetételi stb. lehetőségei. Csak így érhető el, hogy tetszőleges szövegkörnyezetben viszonylag könnyen vissza lehessen adni az adott terminust, szinte az eredeti tükörképeként, és ne fordulhasson elő, hogy bizonyos helyzetekben nehézkessé válik vagy teljesen ellehetetlenül a fordítás.

Folytathatnánk a problémák felsorakoztatását, vagyis az általános kép nem éppen szívderítő. Szinte már csodálkozunk rajta, hogy e látszólag teljes terminológiai zűrzavar ellenére mégis létrejöhet a megértés a szakma művelői között…

Hogyan juthatnánk előbbre?

Univerzális, mindenütt alkalmazható, sőt egyáltalán tökéletes megoldást adni a terminológiai problémákra nyilván nem lehet. Pedig valamilyen megoldásra állandóan szükség van. És azt se felejtsük el, hogy ez a gyökerében megoldhatatlannak látszó probléma számtalan embert érint, a „mezei felhasználóktól" egészen a számítástechnikai guruikig.

De talán éppen a közös gondolkodás segíthet abban, hogy megtaláljuk a módját, mit tehetünk annak érdekében, hogy segítsünk a helyzeten.

1. A legnyilvánvalóbbnak az látszana, hogy gyűjtsük össze azokat a szétszórtan felbukkanó, használhatónak vélt megoldásokat, amelyek könyvekben, folyóiratokban, szakmai anyagokban felmerülnek, és ne hagyjuk őket elfelejtődni. Ahol csak lehetséges, az adatgyűjtésben szerepeljen a forrás, valamint eredeti, idegen nyelvi terminus! Időnként jól használható fogalomgyűjtemény található némelyik szakkönyv függelékeként. Ilyenekkel találkoztam például a Panem kiadó több kiadványában (így S. Schatt: Hogyan működik az ATM?, Ila László: CD — Kompaktlemez, C. Lewart: Modemek c. könyvében) és a ComputerBooks néhány szerzőjénél (például Lengyel Veronika: Az Internet világa c. könyvében).

2. Fel kellene dolgozni a nyomtatásban megjelent, valamint az Interneten



elérhető szó- és kifejezésgyűjteményeket, értelmezéseket, fogalomtárakat. Jó kiindulásnak látszanak magyar nyelven az LSI oktatóközpontja által kiadott terminológiai kiadványok, anyagukat azonban jelentősen ki kellene bővíteni és fel kellene frissíteni. (Lásd e cikk végén a felsorolást.)

3. A gyűjtést és a rendelkezésre álló források feldolgozását is szervezetten kellene végezni. Személy szerint én legjobb megoldásnak azt találnám, ha egy-egy egyetemi vagy főiskolai tanszék (vagy évfolyam) vállalná fel a számítástechnika valamelyik szűkebb területének a gondozását, a terminológiai adatgyűjtést és a gyűjtött anyag feldolgozását. Pontosabban: még jobb lenne, ha párhuzamosan két helyen gondoznának minden szakterületet, és időről időre elküldenék egymásnak, majd egyesítenék anyagaikat.

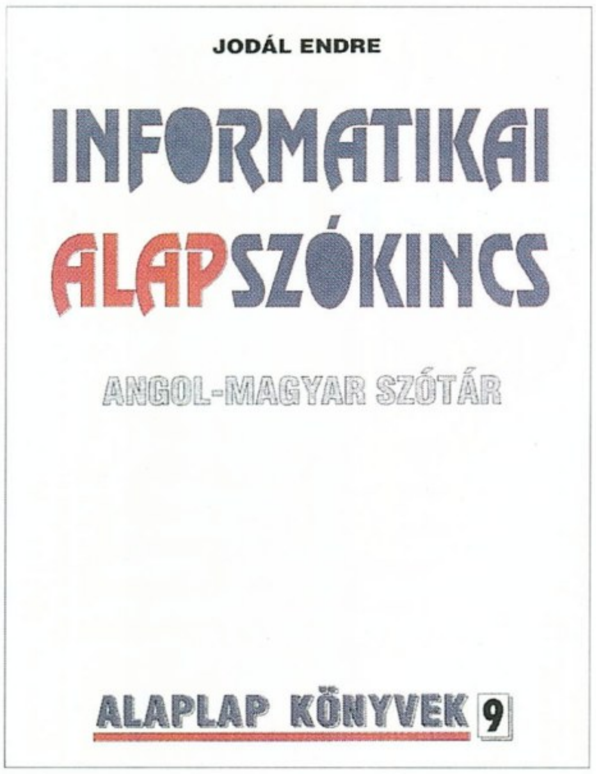
4. Fel kellene venni a kapcsolatot a szakmai kiadókkal és egyes szerzőkkel is, hogy könyveik anyagát a megjelenéssel lehetőleg egy időben bocsássák géppel olvasható formában az illetékes tanszék rendelkezésére. Cserében — amennyiben erre megérnek a feltételek — fel lehetne ajánlani a közreműködést a gépi indexelésben.

5. A gyűjtést valamilyen jól kezelhető hipertextes adatbázisban lenne érdemes végezni, sok kereszthivatkozás beépítésével. Ennek kidolgozását (vagy valamilyen működő rendszer adaptálását) ugyancsak valamelyik egyetemi vagy főiskolai tanszék vállalhatná magára. A munkát úgy kellene megszervezni, hogy szervesen beépülhessen a tanszék hivatalosan elfogadott munkatervébe, a szakmailag értékelhető eredményeket a hallgatók diákköri vagy évfolyami dolgozataként, esetleg más hasonló formában kellene elbírálni.

6. Amennyiben megfelelő mennyiségű és minőségű anyag gyűlne össze, érdemes lenne megpróbálkozni az adott szakterület fogalomkincsének tezauruszszerű rendszerezésével, a fogalomrendszer belső összefüggéseinek feltárásával, színezett gráfként való ábrázolásával, valamint olyan keresőrendszer kidolgozásával, amely a fogalmi összefüggéseket is hasznosítani tudja a keresésben.

7. Érdemes lenne külön is feldolgozni a rövidítéseket, megfelelő mutatókkal és rövid magyarázatokkal ellátva.

8. A megfelelően kiérlelt anyagot (de esetleg már a nyers munkaanyagot is) időről időre aktualizált formában hozzáférhetővé kellene tenni az Új Alaplap CD-mellékletén, a hozzá tartozó hipertextes keresőrendszerrel együtt.



9. A különböző szakterületekről összegyűjtött anyagot meghatározott időnként egyetlen közös rendszerben kellene egyesíteni, és valamilyen alapítvány támogatásával nyomtatott formában is hozzáférhetővé tenni, CD-melléklettel együtt.

10. Feltehetőleg a MorphoLogic és a többi nyelvi műhely fejlesztőgárdája is közreműködne a meginduló munkában, és szakmai tanácsokkal, esetleg más módon is segítséget tudna nyújtani.

Néhány szó a kiadványokról

Lehet, hogy bizonyos mértékig nyitott kapukat döngetek, hiszen az LSI Oktatóközpont Alapítvány szoros kapcsolatokat tart fenn a Gábor Dénes Műszaki Főiskolával, és Kovács Magda fentebb említett munkái arra mutatnak, hogy a főiskolán már évekkel ezelőtt elindult valami hasonló jellegű (bár nem országos) munka a számítástechnikai terminológia összegyűjtésére és rendszerezésére. A feladat azonban nagyobbnak bizonyult, mint kezdetben képzelték. Ők bizonyára sok hasznos tapasztalatot szereztek az eddigiek során, és minden bizonnyal szívesen részt vennének egy országossá bővített mozgalomban.

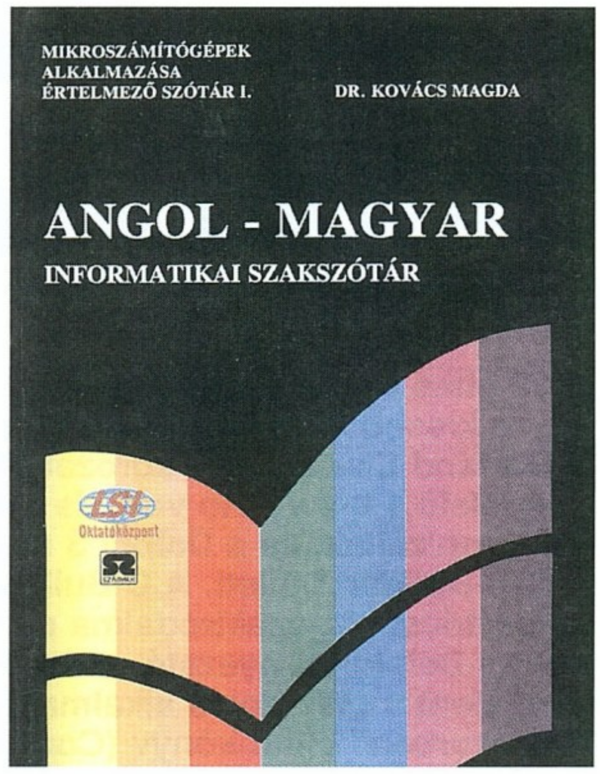
A) „A Mikroszámítógépek Alkalmazása Alapítvány" keretében elkészült, illetve tervezett terminológiai kiadványok. A koncepció a következő kiadványok elkészítését tűzte ki célul:

I. Angol–magyar informatikai szakszótár. Ennek számomra ismert bővített, átdolgozott kiadása 1997-ben jelent meg. A szerző Kovács Magdán kívül a munkában alkotóként is részt vett 5 lektor, 1 szuperlektor, 4 szerkesztő, 6 technikai szerkesztő és 11 további közreműködő. A feldolgozott anyag közel 12 000 címszó.

II. Mikroszámítógépek alkalmazása, Értelmező szótár, I. és II. kötet. Ez angolul és magyarul tartalmazza az elnevezéseket. A feldolgozott anyag szinonimákkal együtt közel 20 000 címszó. Sajnos ennek csak 1991-es kiadását ismerem. Helyenként igen részletes értelmezéseket, magyarázatokat tartalmaz, főleg a hardver vonatkozásában. Hasznos anyag, de jelentős bővítésre és frissítésre szorul.

III. Ötnyelvű (angol-magyar-német-spanyol-eszperantó) informatikai szakszótár. (Ma már nem kapható, de könyvtárakban még hozzáférhető.)

IV. Háromnyelvű (magyar-angol-német) szakszótár, magyar kiinduló nyelvvel. (Korlátozott számban még kapható.)



Eredetileg tervbe vették más nyelvek (például orosz, kínai) bevonását is, ezek a munkálatok azonban tudomásom szerint nem jutottak túl az előkészítés stádiumán.

B) Az „Alaplap Könyvek" sorozatban megjelent terminológiai munkák:

1. Számítástechnikai Alaplexikon I. Jodál Endre: Általános fogalmak. Angol megfelelőkkel és ábécérendes angol visszakeresővel. (Alaplap Könyvek, 5.)

2. Számítástechnikai Alaplexikon II. Jodál Endre: Adatkommunikáció és számítógép-hálózatok. Angol megfelelőkkel és ábécérendes angol visszakeresővel. (Alaplap Könyvek, 6.)

3. Farkas Ernő — Csórián Sándor: PC-szótár. A címszavakhoz értelmező magyarázatok, az angol terminusok feltüntetésével. (Alaplap Könyvek, 7.)

Vargha Dénes

Szakirodalom a fejlesztőknek

Delphi után C++ Builder

A 32 bites Delphi fejlesztőeszköz újabb verzióit egyre kisebb késéssel követte azok C++ nyelvű átirata. A Borland C++ Builder sorozat első tagja a Delphi 2-nek megfelelő 1.0-s verzió volt, de most annak verziószámát szinkronizálták, és a Delphi 3 megfelelője már a C++ Builder 3.0 lett. A C Builder programozási fejlesztőeszköz szakirodalma nem olyan gazdag, mint a Delphi környezeté, ezért örvendetes, hogy most megjelent a „Windows alkalmazások fejlesztése C++ Builder 3 rendszerben" című könyv (ComputerBooks, 1999, 470 oldal). Szerzőgárdája jelentős részben átfedi a Delphi 3-ról szóló könyvét (Benkő Tiborné, Benkő László, Kuzmina Jekatyerina és Tamás Péter).

A Delphi 3 kézikönyvet korábban már ismertettük (Új Alaplap, 1999. október). A C++-os kiadvány szerkezetében hasonló ahhoz, de több helyen tartalmaz sajátosságokat. A könyvbe belelapozva rögtön megtaláljuk az objektumorientált C++ programozás rövid áttekintését. E fejezetnek értékes része a Borland fejlesztőeszközeiben megvalósított komponenskönyvtárral (VCL) végzett munka jellegzetességeinek áttekintése, például, hogy a VCL nem támogatja a többszörös öröklődést, aminek oka, hogy a Borland a C++ Builder kialakításakor igyekezett megteremteni a kompatibilitást a Delphivel. Ez olyan mértékű, hogy az ezzel a fejlesztőeszközzel készült komponensek többségének a kódja Object Pascal, és a C++ Builder gondoskodik a Pascal-kód átfordításáról az új eszközbe. Ennek köszönhetően a fejlesztők előtt nyitva áll az átjárási lehetőség a Delphiből a C++ irányába.

Akik a két fejlesztőrendszerrel párhuzamosan dolgoznak, vagy nem a beépített konverziót választják, azoknak ajánlható a könyv függelékei közül egy összehasonlító táblázat, amely a két fejlesztőeszközben található változótípusok megfeleltetését tartalmazza. Azonban a könyv az elmondottak ellenére nem elsősorban a „Delphiből érkezőknek" készült. Tartalmazza a fejlesztőeszköz rövid ismertetését, és a különböző szintű fejlesztési típusfeladatok megoldási lehetőségeit. A fejezetek anyagában feldolgozott mintakódokat nem kell sajátkezűleg begépelni, mert a korábbi művekhez hasonlóan azokat a CD-melléklet tartalmazza.

A fejezetek sorrendjét követve az egyszerű, egyablakos alkalmazás elkészítésével ismerkedhetünk meg először, és azt később menüvel is kiegészíthetjük. Utána eljuthatunk az egy keretablakban több dokumentumot kezelő MDI-alkalmazások alapjaihoz. A Windows rendszerébe illeszkedő programoknak azonban az egyszerű üzenetfeldolgozáson kívül objektumátadási szinten is együtt kell működniük egymással, ahogy például a Word külön objektumként képes ábrák vagy Excel munkalapok beemelésére.



Az alkalmazások közötti kapcsolatoknak a könyv külön fejezetet szentel, bár jelentőségéhez képest szerény terjedelemben. Tekintve, hogy ennek a kapcsolatrendszernek az ismertetése — a vágólaphasználattól az OLE alapú fogd-és-vidd megvalósításáig — külön könyvet is megtöltene, az általános ismertetésen belül arányosnak mondhatjuk a fejezetet. E témakör jelentőségét tovább növeli, hogy a Windows alkalmazásokban egyre jobban tért hódító ActiveX technológiák megvalósításának is ez az alapja.

A gyárilag kapott, vásárolt vagy éppen a Windowsban már meglévő vezérlőket a C++ Builderrel kiegészítve lehetőségünk van ActiveX vezérlők készítésére is. Ennek leírása a saját komponensek készítését ismertető részben található. Ha pedig (akár a CD-re feltett példák nyomán) elkészültek az új komponensek, a könyvben található leírást követve nem fog gondot okozni azok elhelyezése a komponenspalettán. A gyári és a legteljesebb készletet tartalmazó kliens/szerver verzióban található komponenspaletta ismertetése szintén a könyv függelékei között található.

A Delphi 3-ról szóló műhöz hasonlóan sajnos ez a könyv is szinte már az új C++ Builder verzió (4.0) megjelenésekor került forgalomba. Ilyen komplex fejlesztőeszköz esetében azonban (és tekintettel a verziók közötti átjárhatóságra) valószínűleg mégis értékes olvasmány lesz azoknak, akik vele dolgoznak.

Simay Endre István

ELŐFIZETÉS

Az 1999/......... számtól kezdődően előfizetem

az Új Alaplap című CD-mellékletes havi számítástechnikai folyóiratot

...... példányban ❑ 1 évre, ❑ 1/2 évre.
Az éves előfizetési díj: 6480,– Ft (Ez az összeg az áfát is tartalmazza.)

❑ **Számlát** kérek (banki átutalással fizetek). ❑ **Befizetési csekket** kérek.

Név: ....

(Cég:) ....

Cím: ....

Irányítószám, helység: ....

Dátum:

/aláírás/

APRÓHIRDETÉS

Kérem, hogy az Új Alaplap következő számának Mikrobazár rovatában az alábbi szövegű apróhirdetést jelentessék meg. (A túloldalon ismertetett feltételeket tudomásul veszem.)

....

....

....

....

....

....

....

....

(Maximális terjedelem 300 betű.)

INFORMÁCIÓKÉRÉS

Az Új Alaplap mostani számában megjelent hirdetések közül az általam itt megjelölt kódszámúakhoz részletesebb információt kérek a hirdető cégektől.

01	02	03	04	05	06	07	08	09	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96

Új Alaplap, 1999. júniusi szám. Beküldési határidő: 1999. június 30.

Belföldön díjmentesen is feladható

ÚJ ALAPLAP

VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest

Feladáskor kérjük bérmentesíteni!

FELADÓ:

Név:...

Cím:...

Helység: ....

Irányítószám:..

Telefon: ....

❑ A hirdetés egyéni és egyedi jellegű, ezért kérem ingyenes megjelentetését. Kijelentem, hogy annak tartalma nem sérti senki szerzői jogát.

❑ A hirdetés kereskedelmi célt szolgál. Mellékelem a soronként (60 karakterenként) 300 forintnak megfelelő összeg átutalásáról az igazoló szelvény másolatát. A címzett: Új Alaplap, 1539 Budapest, Pf. 571, illetve átutalásnál az OTP 11706016-20788599 számlaszámra.

- - - - - - - - - - - - - - - - - - -
/aláírás/

Bélyeg helye

ÚJ ALAPLAP

VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest

FELADÓ:

Feladáskor kérjük bérmentesíteni!

a) EGYÉNI

Név: ....

Cím: ....

Helység, ir.sz.: ....

b) CÉGES

Név: ....

Cég: ....

Cím: ....

Helység, ir.sz.: ....

Telefon: ....

- - - - - - - - - - - - - - - - - - -
/aláírás/

Bélyeg helye

ÚJ ALAPLAP

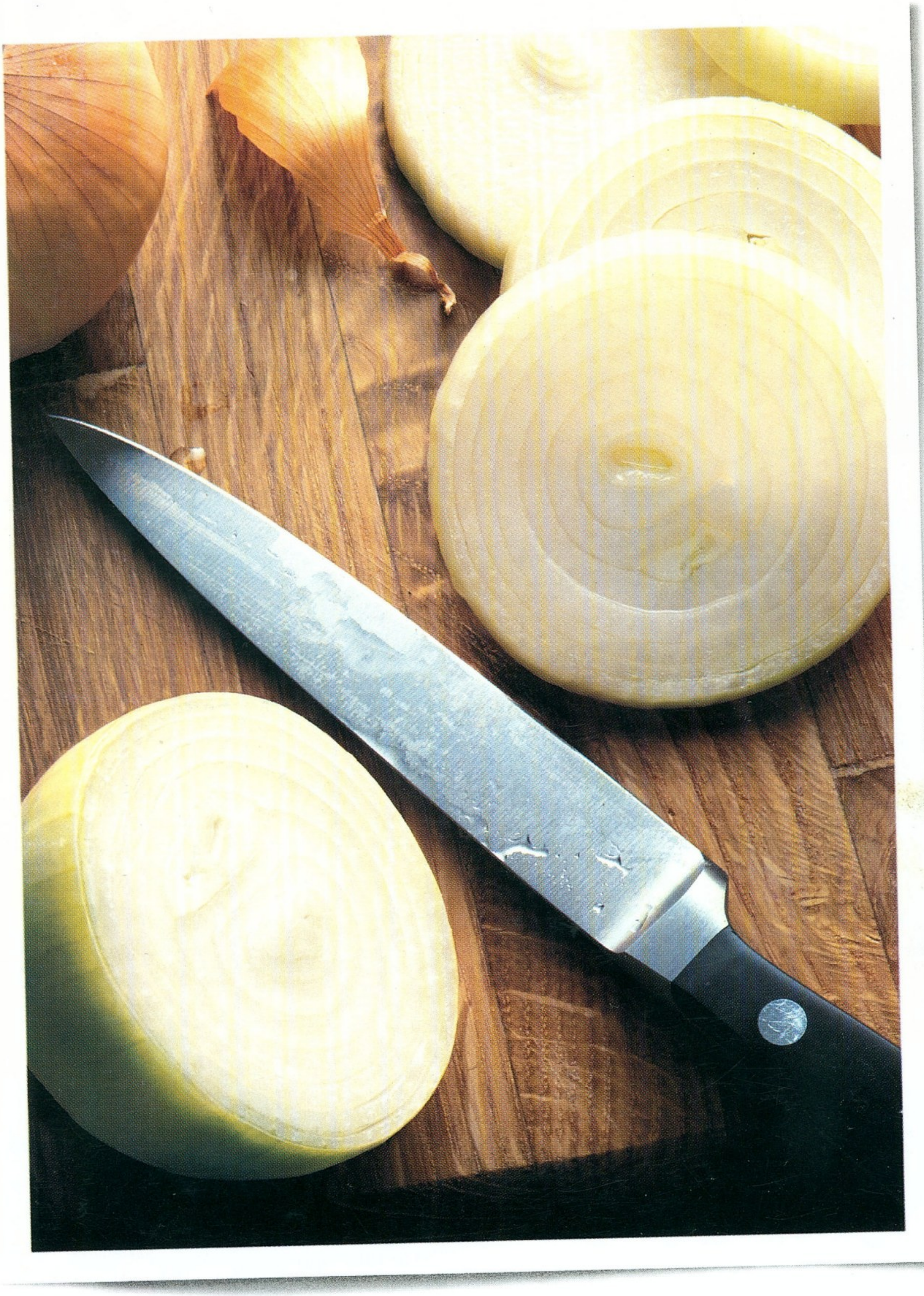
VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest



82
84
88
92
99
2000
1982-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-2000-01-02
VTCD VIDEOTON
Kompaktlemez-gyártó Kft.
Székesfehérvár
Mastered and manufactured by VTCD
IFPI L201
CD-AUDIO
CD-TEXT
CD-EXTRA
CD-ROM
CD-ROM/XA
CD-I
PHOTO-CD
VIDEO-CD
Ø 80mm
Ø 120mm
Kompakt technológia
Kompaktlemez
Kompakt szolgáltatás
Email: vtcd@datanet.hu
Internet: www.vtcd.hu
Tel.:(06-22) 329-132
Fax: (06-22) 329-133
VTCD VIDEOTON
Kompaktlemez-gyártó Kft.
a Videoton csoport tagja
8001 Székesfehérvár,
Pf.:175.
V T C D

HP Vevőszolgálat: 343-0310 http://www.hp.hu/periferiak/

Akció!
Most minden HP PhotoREt II-e
nyomtatóhoz nettó 89.900 Ft-ért
kaphat egy 1 megapixeles HP C30
digitális fényképezőgépet.





MEGTÉVESZTŐEN VALÓSÁGHŰ.



Ha ez a hagyma könnyeket csal a szemébe, annak az az oka, hogy a képet a világ jelenlegi legfejlettebb tintasugaras nyomtatási eljárásával állították elő – a HP forradalmian új PhotoREt II színrétegező technológiájával. Ez az a technikai megoldás, amely lehetővé teszi nyomtatóink számára, hogy akár 16 miniatűr tintacseppet juttassanak el minden parányi képpontba. Az eredmény: lenyűgöző minőségű, fotórealisztikus képek – anélkül, hogy mindez a sebesség rovására menne. Egy szó mint száz, bármennyire siet is, képei ezután mindig fotóminőségűek lesznek. Olyanok, amiket semmi más nem múlhat felül. Kivéve persze magát a valóságot.

HP DESKJET NYOMTATÓK
HP PhotoREt II technológiával

HEWLETT PACKARD
Expanding Possibilities

INFORMÁCIÓKÉRÉS: 11 ▲